

NASA Contractor Report 4316

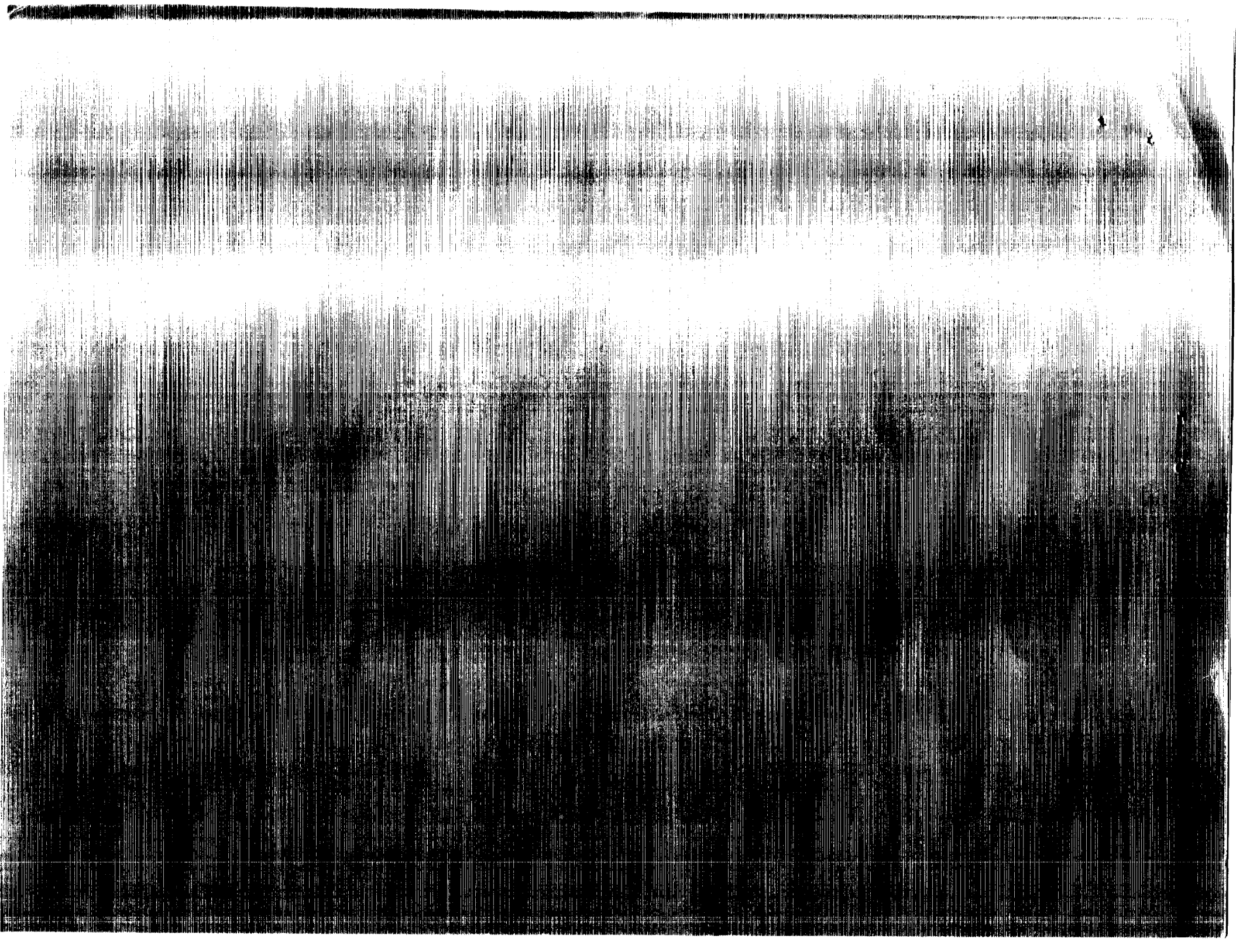
A Knowledge-Based System  
Design/Information Tool

James G. Allen and Scott E. Sikora

CONTRACT NAS9-12451  
AUGUST 1989

(NASA-CR-4316) A KNOWLEDGE-BASED SYSTEM  
DESIGN/INFORMATION TOOL (CRADDER (Charles  
Stark) Lab.) 192 p  
CSCL 129

Unclas  
H1/64 0305032



NASA Contractor Report 4316

# A Knowledge-Based System Design/Information Tool

James G. Allen and Scott E. Sikora  
*Charles Stark Draper Laboratory, Inc.*  
*Cambridge, Massachusetts*

Prepared for  
Ames Research Center  
Dryden Flight Research Facility  
under Contract NAS2-12451



National Aeronautics and  
Space Administration  
Office of Management  
Scientific and Technical  
Information Division

1990





## Table of Contents

1.0 Introduction .....	1
2.0 KCS Overview .....	1
2.1 Problem Domain .....	1
2.2 Requirements.....	1
2.3 Development Environment .....	2
2.4 Knowledge Representation (KR) .....	3
2.4.1 Structured Analysis Methodology.....	5
2.4.2 Conventional H/W Design Methodology.....	6
2.4.3 Rule-Based Models .....	7
3.0 Knowledge Capture Concepts.....	1
3.1 Top Down Decomposition .....	1
3.1.1 Authoring Concepts.....	2
3.1.1.1 Structured Analysis Authoring Methods.....	2
3.1.1.2 H/W Design Authoring Methods .....	7
3.1.2 Browsing Concepts .....	15
3.1.2.1 Structured Analysis Browsing Methods.....	15
3.1.2.2 H/W Design Browsing Methods .....	35
3.1.3 Validation Concepts .....	49
3.2 Bottom Up Schematic Capture.....	55
3.3 Rule-Based Models .....	61
3.3.1 Exploratory Behavioral Models .....	61
3.3.1.1 Operational Models.....	61
3.3.1.2 Failure Mode Models .....	62
3.3.2 Behavior Definition.....	63
3.3.2.1 Rule Capture.....	64
3.3.2.2 Library Objects.....	69
3.3.3 Other Options .....	70
3.4 The Supporting Technologies .....	71
4.0 The Semantic Network.....	1
4.1 The System Design Realm .....	3
4.2 The H/W Design Realm .....	16
4.3 The S/W Design Realm.....	27
4.4 The Utilities Design Realm.....	35

5.0 Rule-Based Models .....	1
5.1 Nose Wheel Steering System .....	1
5.1.1 Mode Logic Description.....	2
5.1.2 The Rule-Based Model.....	3
5.1.3 The Dynamic Behavioral Display .....	3
5.1.4 Conclusions .....	6
5.2 Spin Chute Deployment Model.....	9
5.2.1 The H/W Decomposition .....	9
5.2.2 The Rule-based Model .....	20
5.2.3 The FMEA.....	21
5.2.4 The FTA .....	22
5.2.5 Conclusions .....	35
5.3 Emergency Hydraulic System Model .....	36
5.3.1 The H/W Bottom Up Aggregation .....	36
5.3.2 The Rule-Based Model.....	38
5.3.3 The FMEA.....	38
5.3.4 The FTA .....	38
5.3.5 Conclusions .....	39
6.0 Concluding Remarks.....	1
6.1 KCS Rationale.....	1
6.2 Lessons Learned.....	2
6.3 Guidelines for Future Development.....	3
6.3.1 Phase II.....	4
6.3.2 Phase II+.....	4
Bibliography.....	1

## Introduction

### 1.0 Introduction

The objective of this effort has been to develop a ***Knowledge Capture System*** (KCS) for the Integrated Test Facility (ITF) at the Dryden Flight Research Facility (DFRF). The DFRF is a NASA Ames Research Center (ARC) facility. This system has been used to capture the design and implementation information for NASA's high angle-of-attack research vehicle (HARV), a modified F/A-18A. In particular, the KCS has been used to capture specific characteristics of the design of the HARV fly-by-wire (FBW) flight control system (FCS).

The KCS utilizes artificial intelligence (AI) knowledge-based system (KBS) technology. The KCS enables the user to capture the following characteristics of automated systems:

1. The System Design
2. The Hardware (H/W) Design and Implementation
3. The Software (S/W) Design and Implementation
4. The Utilities (Electrical & Hydraulic) Design and Implementation

A generic version of the KCS has been developed which can be used to capture the design information for any automated system. The deliverable items for this project consist of the prototype generic KCS and an application, which captures selected design characteristics of the HARV FCS.



## KCS Overview

### 2.0 KCS Overview

#### **2.1 Problem Domain**

The Dryden Flight Research Facility operates and tests aircraft with complex life-critical automated systems. The DFRF is currently constructing a new test facility, the Integrated Test Facility, to enable them to work more effectively with the increasing amount of automation found on today's aircraft. This *Knowledge Capture System* is intended to assist the ITF personnel in the tasks of designing, testing and maintaining these complex automated systems.

#### **2.2 Requirements**

The requirements for the KCS evolved from the fly-by-wire flight test experience at the DFRF. This experience has been derived from projects which date back to the F-8 fly-by-wire project which utilized Apollo technology. The more recent fly-by-wire projects include the X-29A technology demonstrator aircraft, the AFTI F-16 advanced fighter technology integration program and the HiMAT demonstrator remotely piloted research vehicle.

These requirements include:

1. A system design capability to ease the capture of design information. The system design capability will provide a graphically structured method for designing complex systems. It will help the designer avoid errors and allow the capture of the design information as it is created. Later in the aircraft development this information, in the form of an intelligent documentation system, will provide information to the test engineers.
2. Online documentation of all the information describing an FCS and the relationships between different disciplinary information. This includes H/W, S/W, redundancy management and flight control law disciplines. The test engineer can then easily and graphically see the design information needed to qualify the system, thus avoiding the in-flight consequences of design errors.

## KCS Overview

3. Expert system functions to help analyze the relationships between the disciplines and uncover where unwanted interactions can occur. These functions can be used by designers, as well as test engineers, to assess the system's operations and avoid serious design errors.
4. Ability to perform failure modes and effects analysis (FMEA) on the many design iterations. Currently, FMEA is only performed on the H/W, not on the system as a whole. Because of the time required to perform an FMEA, the FMEA is usually performed once and is done with an early design iteration. The inability to analyze the final design raises questions of the FMEA's value. Automated FMEA using the current online design is one example of a capability that would assist designers and test engineers in finding serious design errors in a timely manner.
5. Links from the system requirements to the S/W and H/W designs. The links will allow the system requirements to be verified against the proposed implementation. Verification could then be done in an automated fashion, prior to committing to the build phase. This rapid prototyping concept would increase the chance of finding serious design errors prior to flight test.

### **2.3 Development Environment**

A rapid prototyping development environment has been used to develop the KCS. The work station is a Symbolics LISP machine. The Intellicorp Knowledge Engineering Environment (KEE™) Software Development System "shell" was used to provide and support several useful paradigms:

- an object oriented language
- rule-based reasoning
- truth maintenance
- multiple worlds
- lisp methods
- graphics
- active values

## KCS Overview

Common LISP was used to code the KCS functions and methods. The software has been designed to be portable to other platforms, e.g. the VAX and the SUN machines.

### 2.4 Knowledge Representation (KR)

The knowledge base consists of a semantic network of objects and of rule-based models which utilize these objects. The semantic network is composed of four realms of knowledge: the system design realm, the hardware design realm, the software design realm and the utilities design realm. Each of these realms is a highly correlated (strongly linked) design dimension within the problem domain and is implemented with linked hierarchical networks of objects. The integrated semantic network is formed by linking the hierarchical networks of the four realms. This knowledge representation scheme was chosen to support a major goal of this KCS, namely that it should provide an *integrated environment*. Here the intent is to integrate the knowledge capture of these four realms within the problem domain.

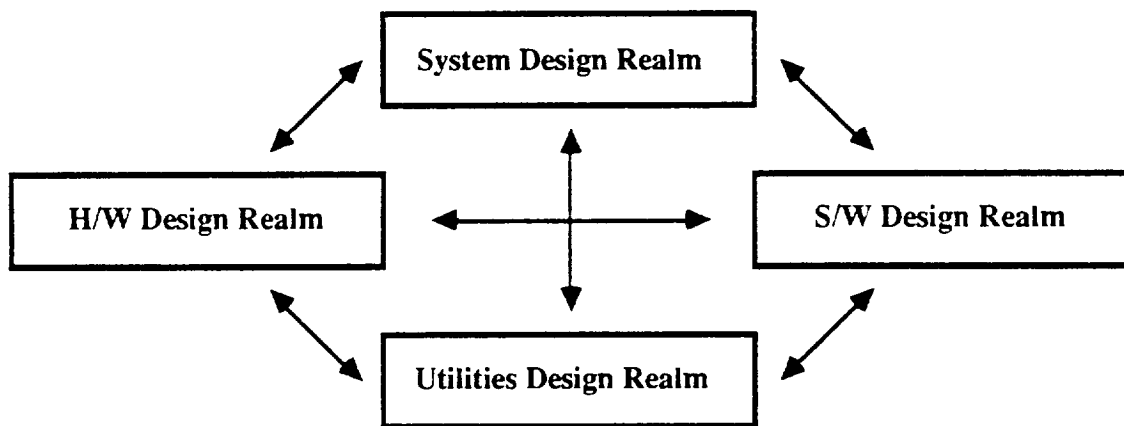


Figure 2-1. Knowledge Realms and Their Linkage

The objects are individually represented with a frame based representation.<sup>1</sup> This representation provides the ability to associate properties with each of the objects and to tailor the appropriate inherited properties and methods to each of the network hierarchies. For example, power consumption is a natural property of H/W objects and memory requirement is a natural property of S/W processes.

## KCS Overview

The KCS includes authoring mechanisms which enable the user to build a network uniquely tailored to a particular FCS, which in this case is an F/A-18A FCS. These authoring mechanisms include decomposition and schematic capture capabilities. Figure 2-2 depicts the notion of how the semantic network is built using authoring mechanisms embedded in a knowledge capture system. The KCS also includes browsing mechanisms which provide access to the semantic network knowledge. Figure 2-2 also depicts the notion that rule-based models are used to perform reasoning on the objects defined in the semantic network.

Rule-based models have been integrated into the KCS. In this case, integration refers to the use of data structures in the frame based representation which are compatible with an inference engine. Thus, it is possible to develop models which utilize the objects defined by the authoring mechanisms. This capability allows the KCS to go significantly beyond the capabilities provided by ordinary structured analysis tools. Namely, the objects defined by the decomposition process can be used in dynamic models which enable the user to test, explore and generally better understand the working of the captured design and implementation knowledge.

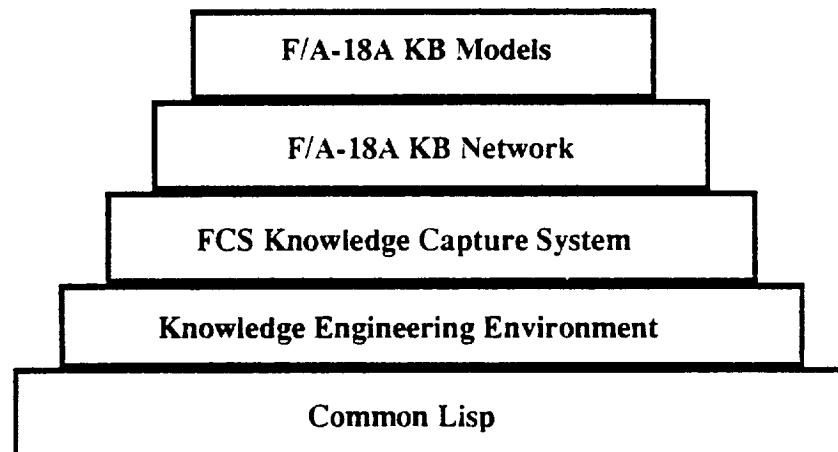


Figure 2-2. The Layered KBS Architecture

A goal, which has guided the selection of the knowledge representation for this KCS, has been to utilize mature techniques and methodologies. The implementation reflects this goal and enforces it through the use of structured authoring mechanisms. The use of



## KCS Overview

structured analysis, networks, frames, data driven models and goal driven models all reflect mature technologies<sup>1 - 6</sup> that date back to the 1960's and 1970's. Conventional H/W design methodology reflects an approach which has evolved since the 1950's and is in use today. The maturity of these techniques and methodologies have provided a stable, coherent and viable infrastructure for the KCS.

### **2.4.1 Structured Analysis Methodology**

The structured analysis (SA) methodology<sup>5, 6</sup> is used in the system design realm and the S/W design realm. The structured analysis methodology is based on a top down hierarchical decomposition of system requirements. This decomposition continues until the requirements are given with an adequate degree of detail. Each node in the resultant tree is called a process and provided with a process description. In addition, external objects and data store objects are identified. The data flow between these objects (from process to process, between processes and externals, between data stores and processes, etc.) is identified in a data dictionary. All of this information is depicted graphically in data flow diagrams (DFDs).

The structured analysis design methodology is utilized here with the understanding that it is a design approach generally considered to be good practise within this problem domain and by NASA. It should be recognized that this KCS is merely integrating this mature and accepted methodology as a proper basis for the KCS KR. It will be seen that this methodology meshes ideally with a mature AI semantic network representation which utilizes hierarchical networks of frame based objects.

Figure 2-3 depicts a level 0 DFD. Figure 2-4 depicts a level 1 DFD, which is an expansion of one of the level 0 DFD processes. The system design requirements shown here are those for the F/A-18A FCS. These are typical DFDs with typical graphic images. The DFDs and their graphic images are used as a graphical front end for the SA methodology objects in the semantic network. The DFD's and their images are mouse sensitive and possess menus for entering and accessing knowledge.

The concepts of a process, an external, a data store and a data flow, as defined by structured analysis, are identified here as objects and individually represented as frames. The properties of the process, external, data store and data flow objects are stored in the slots of the individual frames associated with each of these objects. The nature of the slot

## KCS Overview

values can draw from the full spectrum of the paradigms supported by KEE™. Namely, they may be simple values, pointers to other frames, inherited values, active values, rules, etc. It is intended that the pointers, which are stored as slot values, will provide access to the related hardware, software and utilities implementation knowledge stored elsewhere.

A hierarchical representation scheme is used for each of the four types of objects (processes, externals, data stores and data flows). Each of these four hierarchies forms an individual knowledge base (KB). In each case, the hierarchy is used to allow properties to be inherited and to identify the natural linkage between individual objects. These individual KBs are linked with pointers.

### **2.4.2 Conventional H/W Design Methodology**

The knowledge representation for the hardware design realm and the utilities design realm is based upon the H/W design methodology typically used in this problem domain. The nature of the representation is similar although not identical to the structured analysis methodology. The hardware objects are represented graphically as blocks and these objects are decomposed in a hierarchical fashion until they have been described to an adequate degree of detail. This decomposition is represented graphically with H/W block diagrams.

The connectivity between these H/W objects is generally indicated graphically with lines and arrow heads. These lines may represent specific H/W connection devices such as data buses, a flow of information or a form of control. In any case, this connectivity can be represented in the form of objects of a specific type and may possess a hierarchical characteristic. Here this connectivity is called signal flow (S/F).

The concepts of hardware and signal flow are identified in the KCS as being objects and are individually represented as frames. The properties of these objects are stored in the slots of the individual frames. The nature of the slot values and the hierarchical relationship of the frame representation is the same as that provided for the SA objects.

Figure 2-5 depicts a level 0 H/W diagram. The H/W design top level requirements shown here are those for the F/A-18A FCS. This is a typical H/W diagram with typical graphic images. The H/W diagrams and their images are used as a graphical front end for the H/W

## KCS Overview

design methodology objects in the semantic network. The H/W diagrams and their images are mouse sensitive and possess menus for entering and accessing knowledge.

Figure 2-3 depicts a level 0 DFD. Figure 2-4 depicts a level 1 DFD, which is an expansion of one of the level 0 DFD processes. The system design requirements shown here are those for the F/A-18A FCS. These are typical DFDs with typical graphic images. The DFDs and their graphic images are used as a graphical front end for the SA methodology objects in the semantic network. The DFD's and their images are mouse sensitive and possess menus for entering and accessing knowledge.

In conventional designs, the final products utilize schematics and mechanical drawings for the real world objects. This practise is represented in the KCS through the use of a dual graphical representation scheme which integrates the box and line graphical representation at the top level with bitmaps of the schematics and mechanical drawings at the lower levels. The use of the dual graphical representation is illustrated in Figures 2-6. Note the use of hotspots (shaded areas in the circuit diagram) to link the dual graphical representations.

### **2.4.3 Rule-Based Models**

An environment has been provided which incorporates an inference engine and problem domain objects with compatible data structures. This environment has been supplied to allow the development of dynamic rule-based models which answer such questions as:

- How does this subsystem work?
- What happens if this component fails?
- How redundant is this functionality?
- What functionality exists in this mode of operation?
- What are the memory and throughput associated with a given allocation of S/W processes?

During the development phase, these models are intended to support a rapid prototyping environment. The same models are then intended to support the verification and validation phase. Subsequently, these models are intended to support the operational

## KCS Overview

phase and satisfy tutorial needs. As the control system ages, these same models can be used to support design and system upgrades.

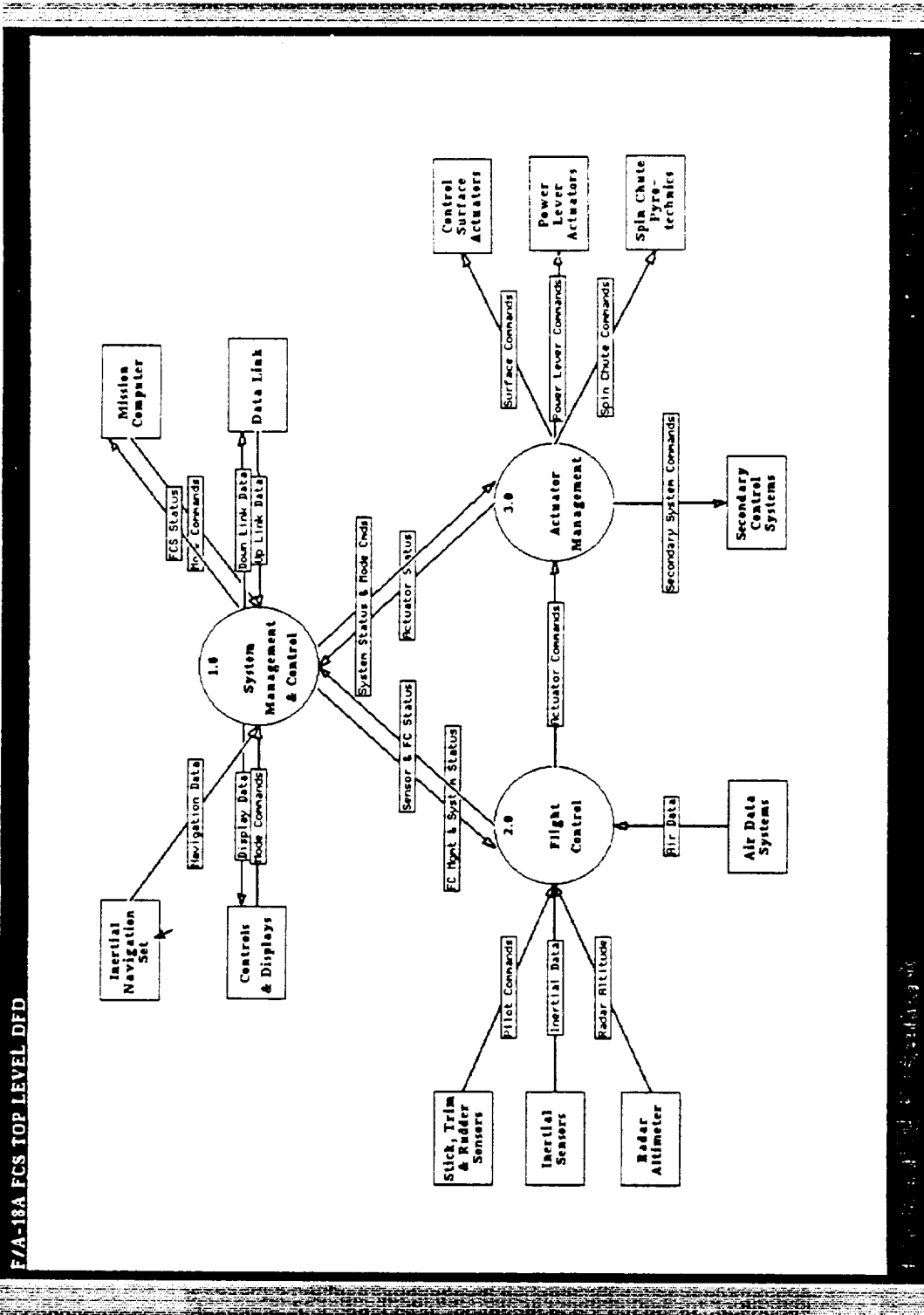


Figure 2-3. F/A-18A Level 0 DFD - The Top Level Data Flow Diagram

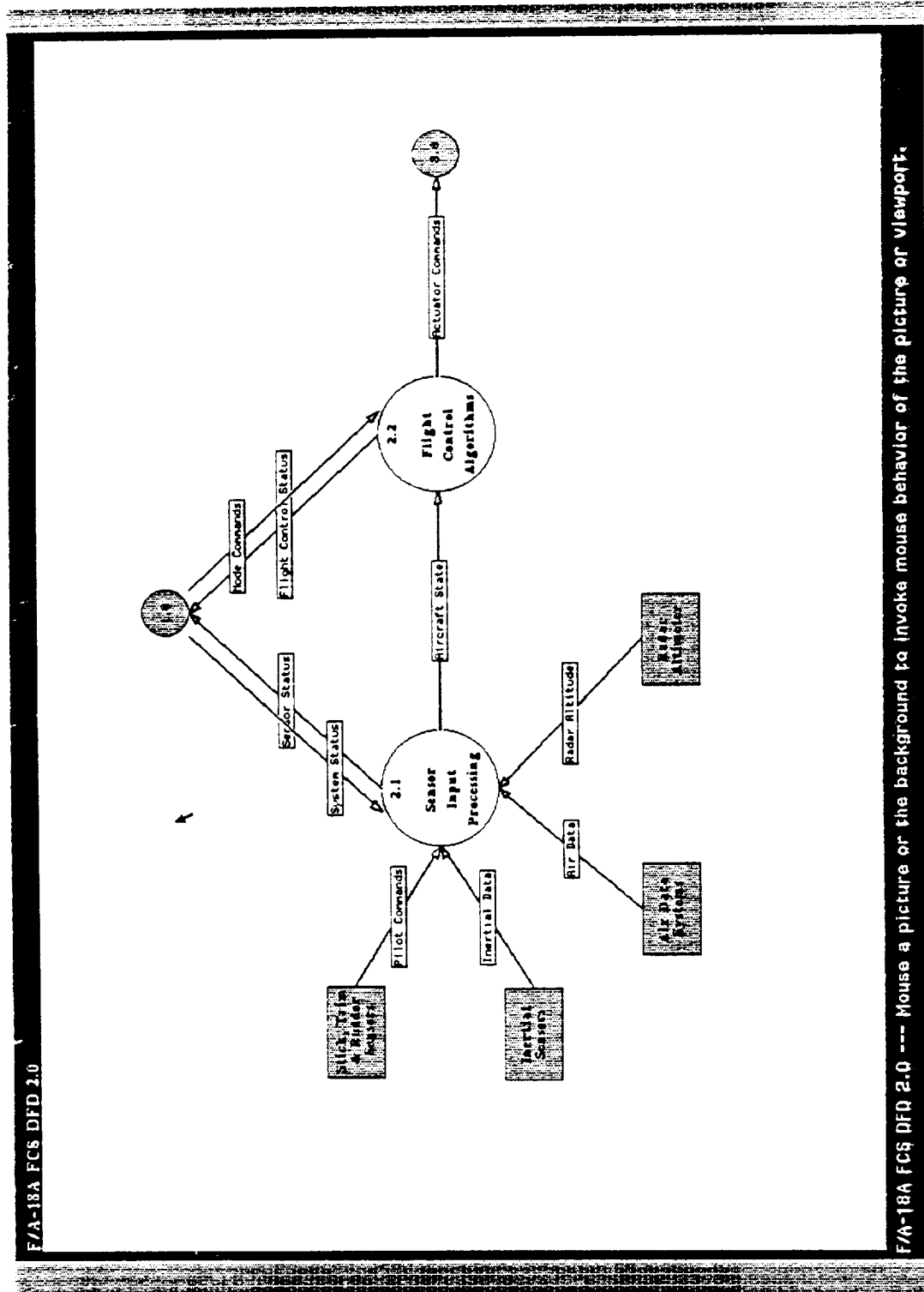


Figure 2-4. F/A-18A Level 1 DFD: DFD 2.0 - Flight Control

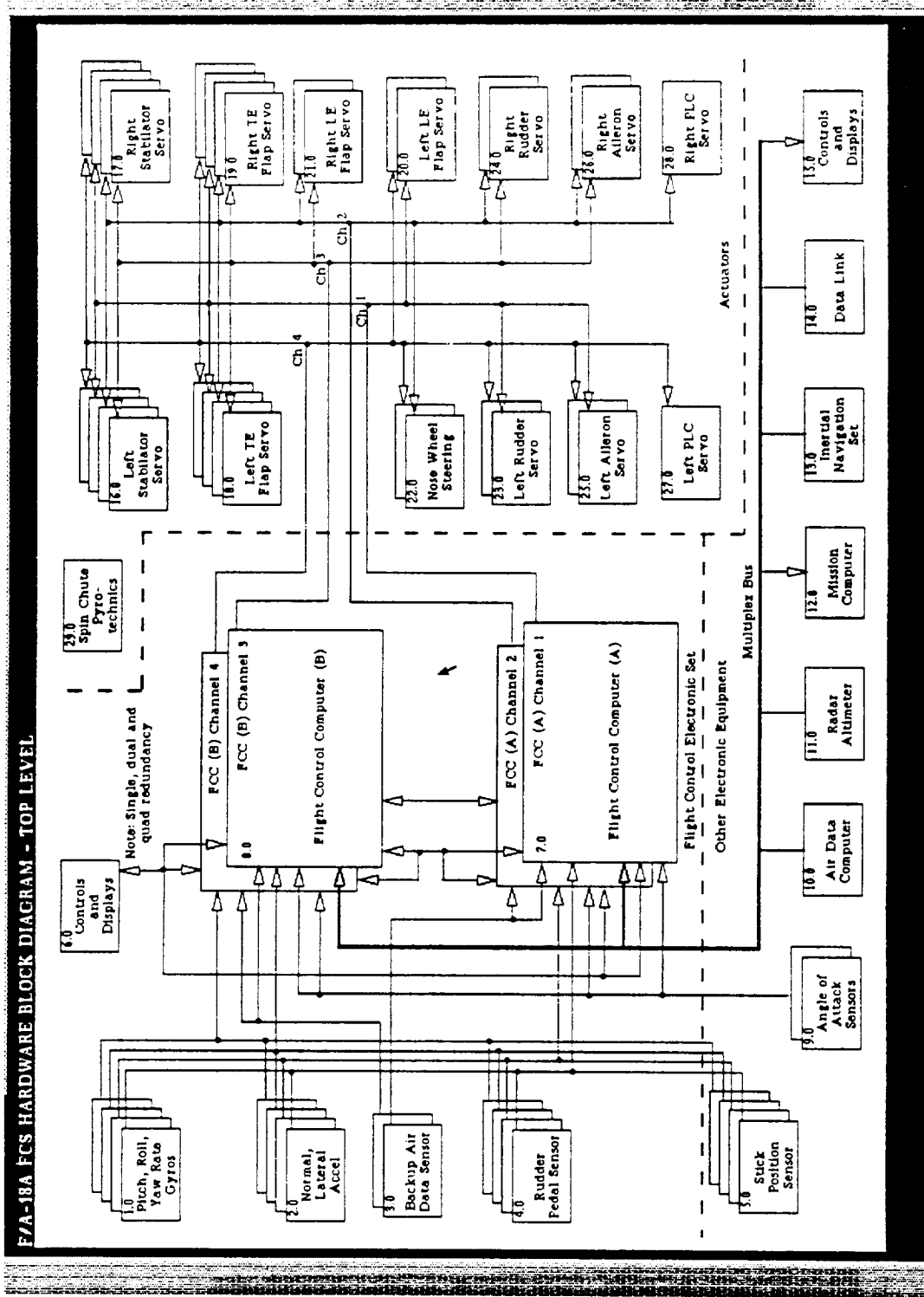


Figure 2-5. F/A-18A Top Level H/W Block Diagram

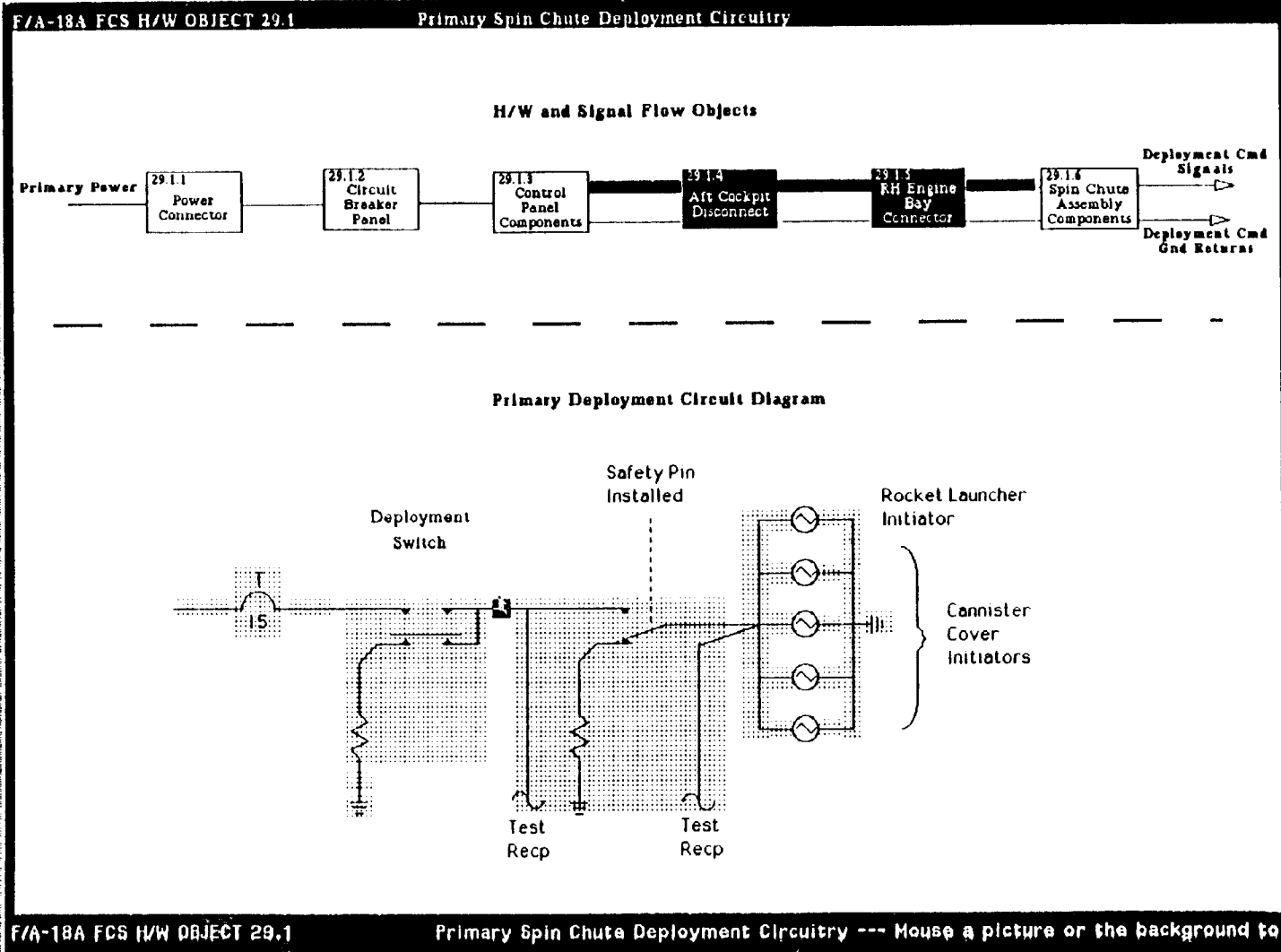


Figure 2-6. F/A-18A Level 2 H/W Diagram - Primary Spin Chute Deployment Circuitry



### **3.0 Knowledge Capture Concepts**

*Authoring mechanisms* are used by the KCS to capture information. These authoring mechanisms constrain the user to utilize the design methodologies which are appropriate to this problem domain. Mouse and menu mechanisms are provided which implement these authoring mechanisms in a user friendly fashion. The *semantic network* implementation is a natural end result of this authoring process.

These authoring mechanisms recognize the use of both top down and bottom up design approaches in the evolution of the final design of an automated system. Top down decomposition methods are provided for both the structured analysis methodology and the H/W design methodology based authoring methods. The bottom up design approach is more commonly found when the H/W designs are developed as electronic schematics and drawings of mechanical components. Schematic capture authoring mechanisms have been incorporated into the system which implement this form of bottom up authoring.

A second and essential part of this system is the ability to browse through the knowledge base to utilize the captured knowledge. The *browsing mechanisms* which have been implemented have been selected to provide an intelligent access to the documented knowledge such that this second dimension of the knowledge based system can be thought of as a form of intelligent documentation. The inherent structure of the semantic networks has been utilized to allow the user to explore the knowledge base along the thinking lines that he/she would normally use when working in this problem domain. When we solve our problems, we explore the solution space in a fashion that is referred to as a form of spreading activation. This theory of spreading activation suggests that we search in ever growing (and related) circles for solutions to our problem. The browsing mechanisms have been designed to support such a search with a structured object linkage that is tailored to the problem domain and integrated across the four design realms.

#### **3.1 Top Down Decomposition**

The top down decomposition knowledge capture is performed with the assistance of methods which implement authoring mechanisms appropriate to the chosen methodologies. The result is a structured knowledge base which is consistent with the chosen

## Knowledge Capture Concepts

methodologies. Browsing mechanisms are provided to give the user access to this knowledge base.

### **3.1.1 Authoring Concepts**

Authoring methods have been created which comply with the type and connectivity constraints imposed by the problem domain methodologies. These constraints, which are imposed by the methodologies, have been implemented in a transparent fashion via a user friendly interface which enforces the user's compliance with these methodologies. These user interface graphics utilize mouse and menu mechanisms in a graphic, diagrammatic format consistent with the problem domain.

While conforming to the constraints of the methodologies, the mechanization has made use of object oriented hierarchical concepts to implement a KCS which can gracefully grow in an incremental fashion. In addition, the KCS has been designed to make use of access to network technology so as to support an open environment. This open environment is intended to enable the KCS to communicate with other platforms on which relevant system knowledge resides, such as CAD systems which define the S/W code and H/W designs. This open environment concept is intended to support the use of multimedia technology.

#### 3.1.1.1 Structured Analysis Authoring Methods

The structured analysis methodology is quite specific in its definition of useful design objects, namely it specifies that designs make use four generic types of objects: data flows, data stores, externals and processes. In addition, this methodology specifies the form of connectivity between these objects. Graphic diagrams, called data flow diagrams (DFDs), which depict these objects and their connectivity, are also an inherent part of this methodology.

This typing and these constraints are reflected directly by the methods which create and delete these objects, and their presentation on the DFDs. Figure 3.1.1-1 depicts the object creation commands provided on the DFD cascading menu. These commands create the graphical representation of the selected type of object and simultaneously create an object in the appropriate hierarchical network within the semantic network. This new object contains the properties of the object such as criticality and its network linkage.

## Knowledge Capture Concepts

Property modification methods are also used to simplify the specification of object properties, while simultaneously constraining the user to enter values which conform to range and type. Figure 3.1.1-2 depicts the process change commands provided on the process cascading menu. Each of the several process property change commands activates further user support when the command is selected. Figure 3.1.1-3 depicts the menu of choices provided when the user chooses to change the process criticality. It constrains the user to select one of the three valid criticalities (flight critical, mission critical or ground critical).

The data flow change commands are similar to those depicted in Figure 3.1.1-2. Figure 3.1.1-4 depicts the menu of choices provided when the user chooses to modify data flow object connectivity. These options allow the user to connect the data flow object to process objects and external objects in the semantic network and to visualize the connection graphically on the DFD. The other options allow the user to modify the data flow stubs that are created when an object is expanded. These options allow the user to decompose the new data flow objects (stubs) by splitting them and to disconnect them during the trial and error phase of authoring.

Object decomposition is controlled to insure that children conform to the connectivity already specified for the parents. When the expand command is selected from the process menu, the new DFD is created and displayed to the user. Figure 3.1.1-5 depicts the process and external interface images and the data flow stub images that are automatically created when Process 1.0 (System Management & Control) is expanded. At the same time that the new DFD graph is created, the new objects are created and appropriately linked into the semantic network. This controlled incremental growth of the semantic network is essential to the maintenance of valid links within the network.

## Knowledge Capture Concepts

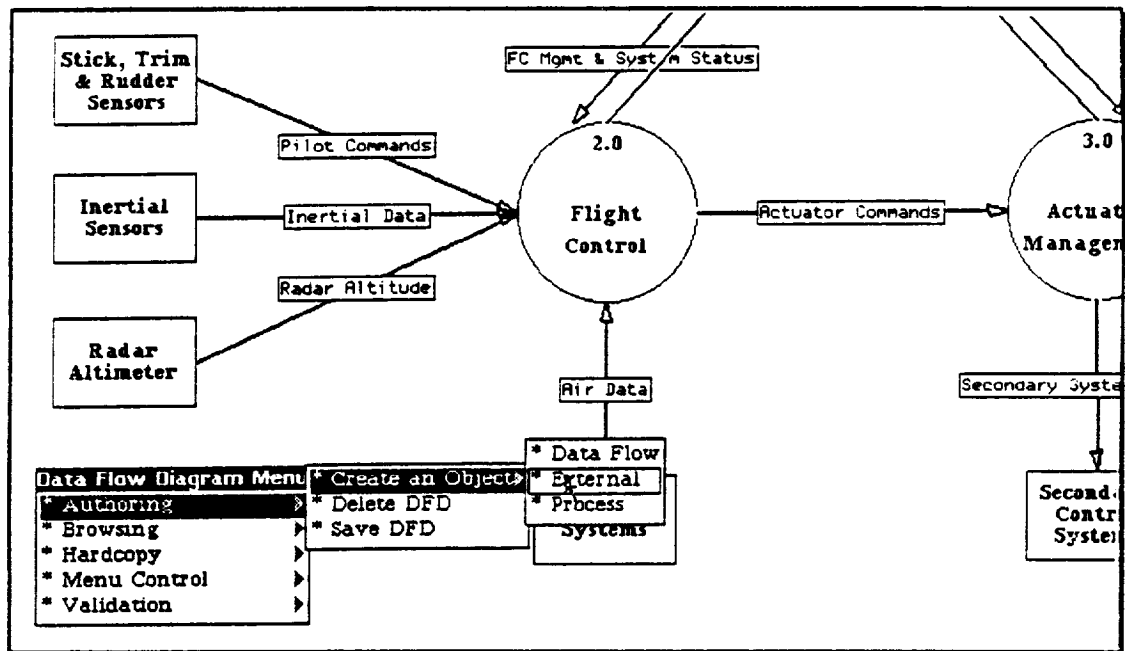


Figure 3.1.1-1. DFD Menu Authoring Options

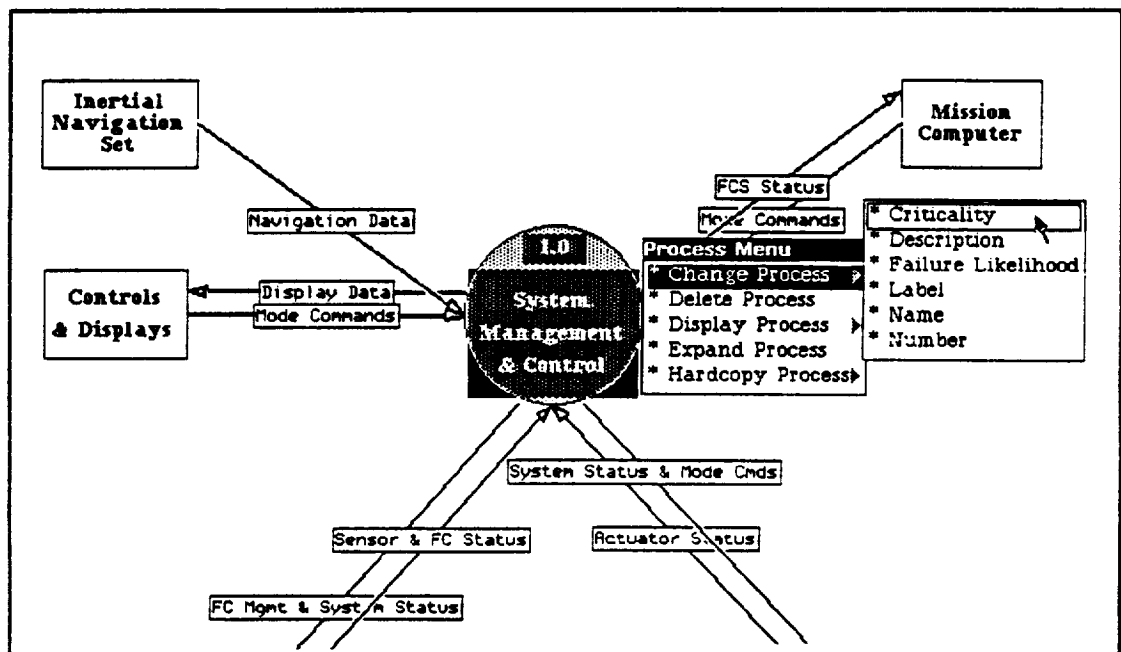


Figure 3.1.1-2. Process Menu Property Change Options

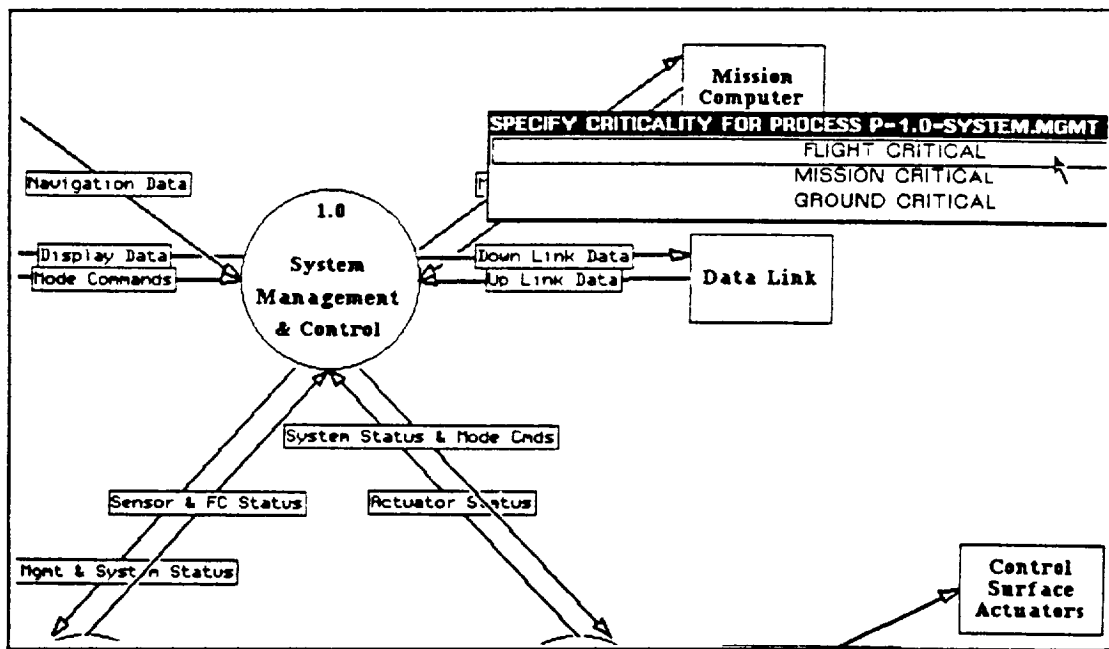


Figure 3.1.1-3. Process Criticality Menu Change Options

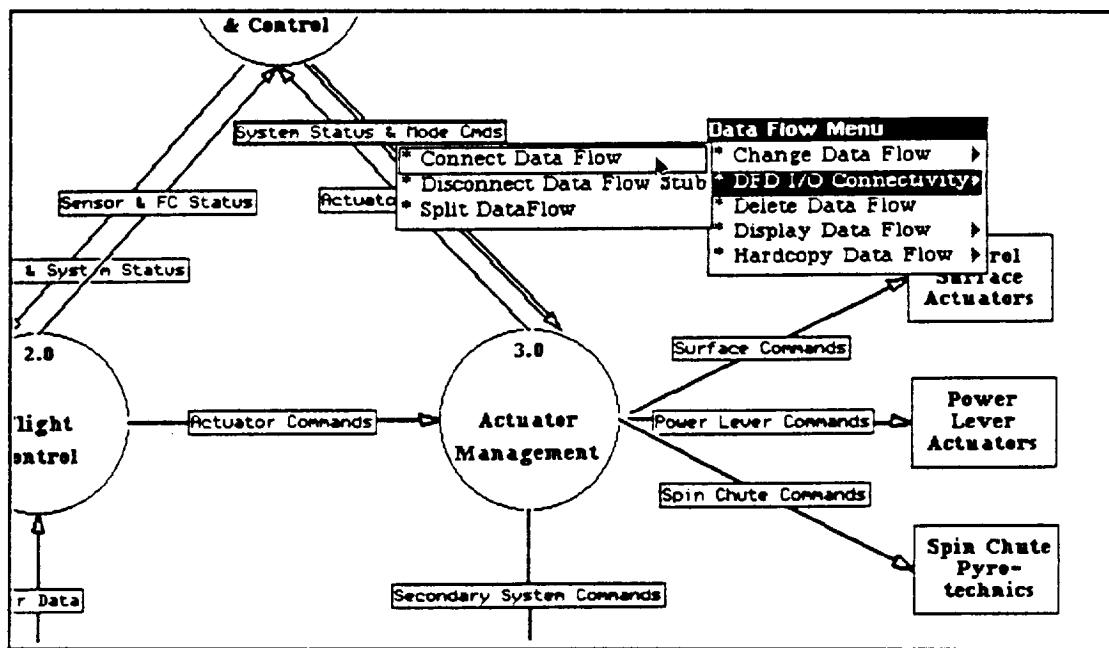


Figure 3.1.1-4. Data Flow Menu Connectivity Options

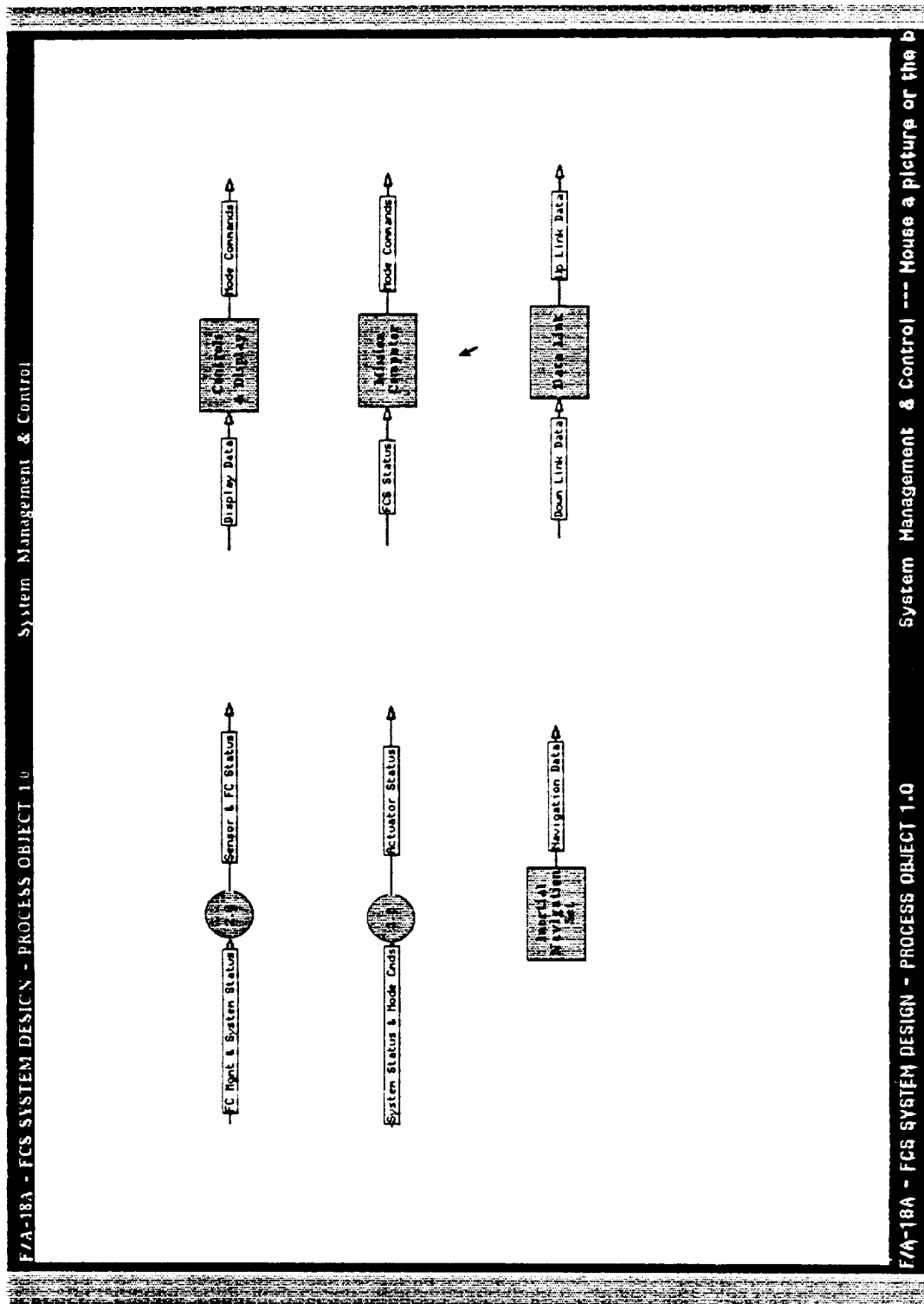


Figure 3.1.1-5. Typical Process Expansion Interface and Data Stub Images

## Knowledge Capture Concepts

### 3.1.1.2 H/W Design Authoring Methods

The H/W design methodology supports multiple abstractions. At the top level, the abstraction is intended to convey the understandings associated with complex objects, such as an entire computer, and information flow, such as commands to the control surfaces. At the bottom level, the abstraction is intended to convey the understandings associated with simple components, such as a resistor, and information flow mechanisms, such as a wire. In between the top level and bottom level, the useful abstractions become a mix of these two extremes of representation.

This methodology has been represented with two fundamental types of objects, the H/W object and the signal flow object. The complexity of the H/W objects ranges from that associated with a computer to the simple component level. The complexity of the signal flow objects ranges from that of abstract information flow through a midrange of a cable representation to that of a wire. Graphic diagrams, called hardware diagrams (H/W diagrams), which depict these objects and their connectivity, are also an inherent part of this methodology.

The structured analysis methodology and the H/W design methodology bear considerable resemblance. Enough so, that it is more instructive to look at their differences. A major difference is that the SA methodology rests on a body of literature<sup>5,6</sup> and conventions which are highly focused and recognized within the problem domain. Conversely, the H/W design methodology merely rests upon conventions which have evolved in practice. This difference is reflected in the diversity of abstractions associated with the H/W design methodology when compared to the SA methodology.

The added complexity of abstraction for the H/W design methodology has dictated the use of more than one graphical representation. The simple box, circle and line graphical representation used for DFDs is thoughtfully adequate and consistent with the SA methodology. Conversely, the simple box and line graphical representation used for the top level H/W diagram fails to support the user interface for H/W diagrams at the mid and lower levels. Drawings, such as electrical diagrams and mechanical drawings, have been added. These drawings, which are necessary at the lower levels, are linked to the box and line representations, which are more appropriate at the top and upper level H/W diagrams.

## Knowledge Capture Concepts

Menu-driven authoring methods are provided for the creation of H/W objects and S/F objects which comply to the typing associated with the H/W Design methodology. Figure 3.1.1-6 depicts the menu provided to the user when a H/W diagram is moused. In the top down decomposition mode, these authoring methods help the user create the allowed types of objects. These commands create the graphical representation of the selected type of object and simultaneously create an object in the appropriate hierarchy within the semantic network. This new object contains the properties of the object such as its criticality and status.

Property modification methods are also used to simplify the specification of object properties, while simultaneously constraining the user to enter values which conform to range and type. Figure 3.1.1-7 depicts the H/W object change commands provided on the H/W object cascading menu. These menu items allow the user to modify the criticality, description, failure likelihood, label, name, number and status. Figure 3.1.1-8 depicts the menu of choices provided when the user chooses to change the status of an object. It constrains the user to select an operational state which is either OK or failed.

Other H/W object authoring capabilities are provided which allow the user to delete, expand, modify the image and specify the replication of a selected object. The image modification and replication options are depicted in Figures 3.1.1-9 and 3.1.1-10.

The signal flow object authoring capabilities change commands are similar to those provided for H/W objects and are depicted in Figure 3.1.1-11. Here the user is given the capability to change the criticality, description, destination(s), failure likelihood, name, sources(s) and status of a selected S/F object. In this case, the sources(s) options are depicted. Figure 3.1.1-12 depicts the multiple selection menu provided for the user when the source deletion option is selected. One or more of the H/W objects displayed on the pop up menu may be selected for deletion. The deletion method then deletes the selected connections in a fashion consistent with the semantic network representation.

As the top level design within the H/W design methodology are decomposed it becomes desirable to use another graphical representation. Figure 3.1.1-13 depicts an example of the use of this dual representation capability. Here a circuit diagram has been added to the box and line diagram associated with a top down decomposition. Hot spots (the shaded areas) have been added to the circuit diagram. These hot spots associate components in the schematic to the appropriate H/W and/or S/F objects.



## Knowledge Capture Concepts

The hot spot options, also depicted in Figure 3.1.1-13, allow the user to integrate the two representations. These authoring methods allow the user to associate a hot spot with the object selected or to delete an existing association. In this case, hot spots for the *aft cockpit disconnect* have already been defined such that its two physical connections are highlighted on the circuit diagram. This highlighting was activated because the *aft cockpit disconnect* H/W object has been selected by the mouse in this particular situation. When each H/W or S/F object is moused on this H/W diagram a similar hot spot linkage will appear along with the menu of authoring and browsing options.

Figure 2-6 depicts the use of hot spots when a hot spot on a diagram is selected with the mouse. In this case, a connection (wire) on the circuit diagram has been moused. Here it can be seen that this connection is physically implemented with several components. These components carry the command in the cockpit to the spin chute control elements located on the tail of the aircraft.

## Knowledge Capture Concepts

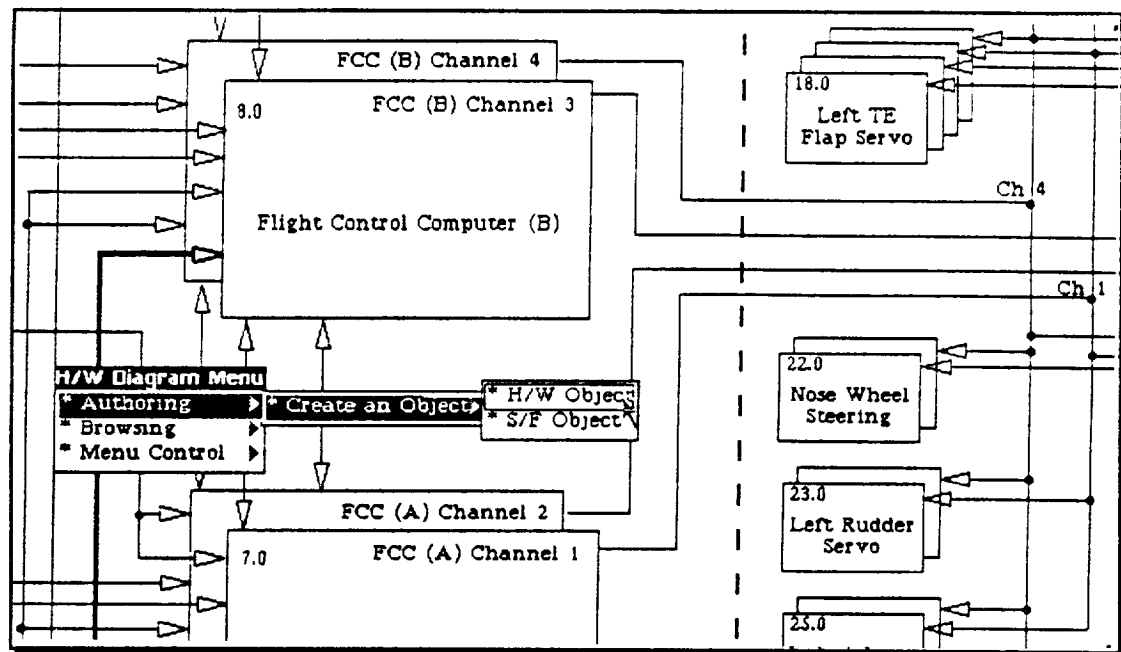


Figure 3.1.1-6. H/W Diagram Menu Authoring Options

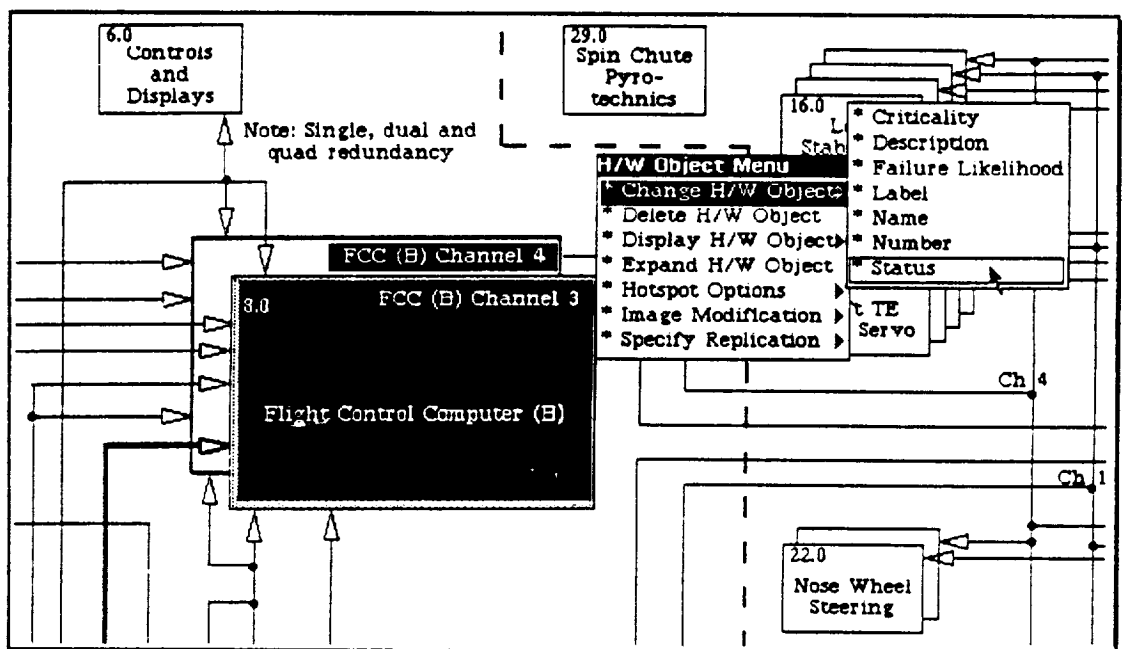


Figure 3.1.1-7. H/W Object Menu Change Options

ORIGINAL PAGE  
OF POOR QUALITY

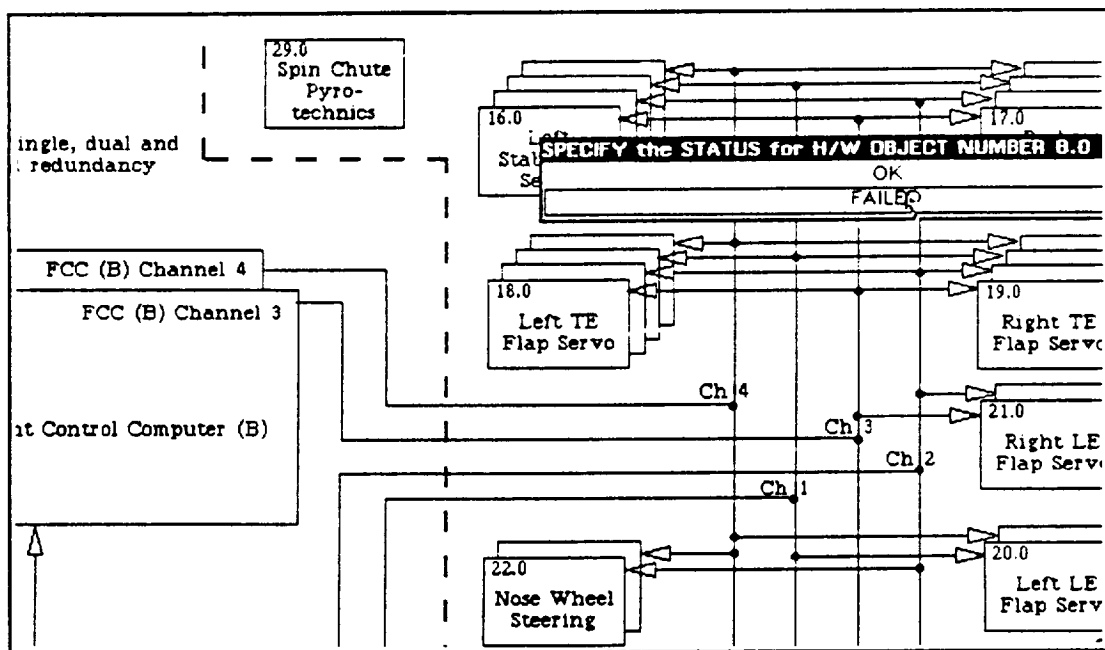


Figure 3.1.1-8. H/W Object Status Selection Options

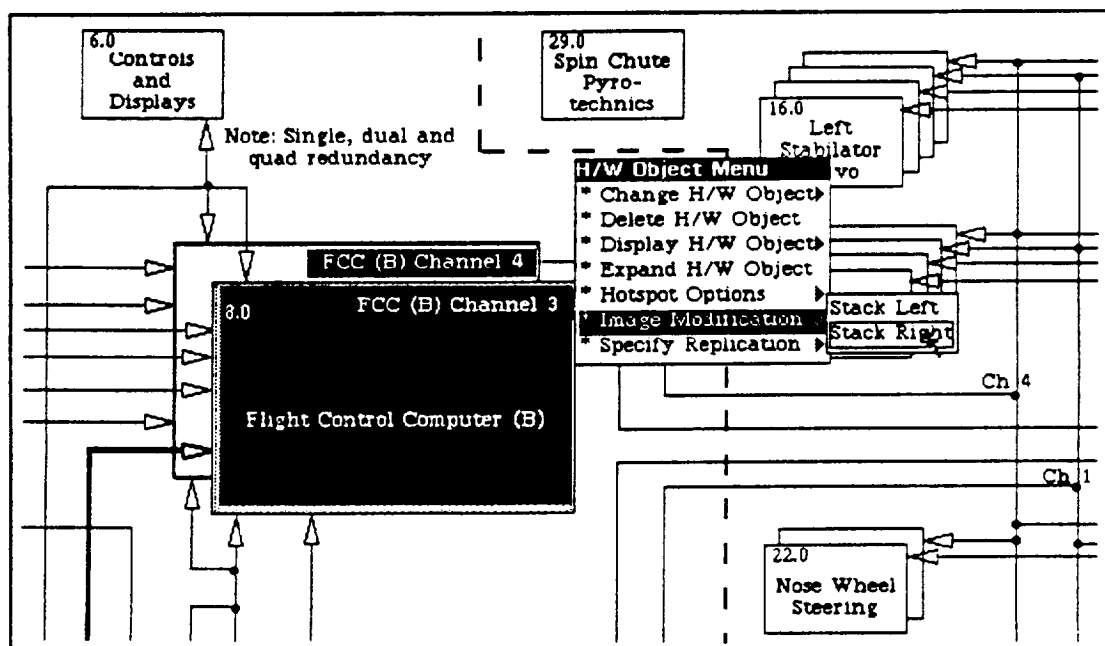


Figure 3.1.1-9. H/W Object Image Modification Authoring Options

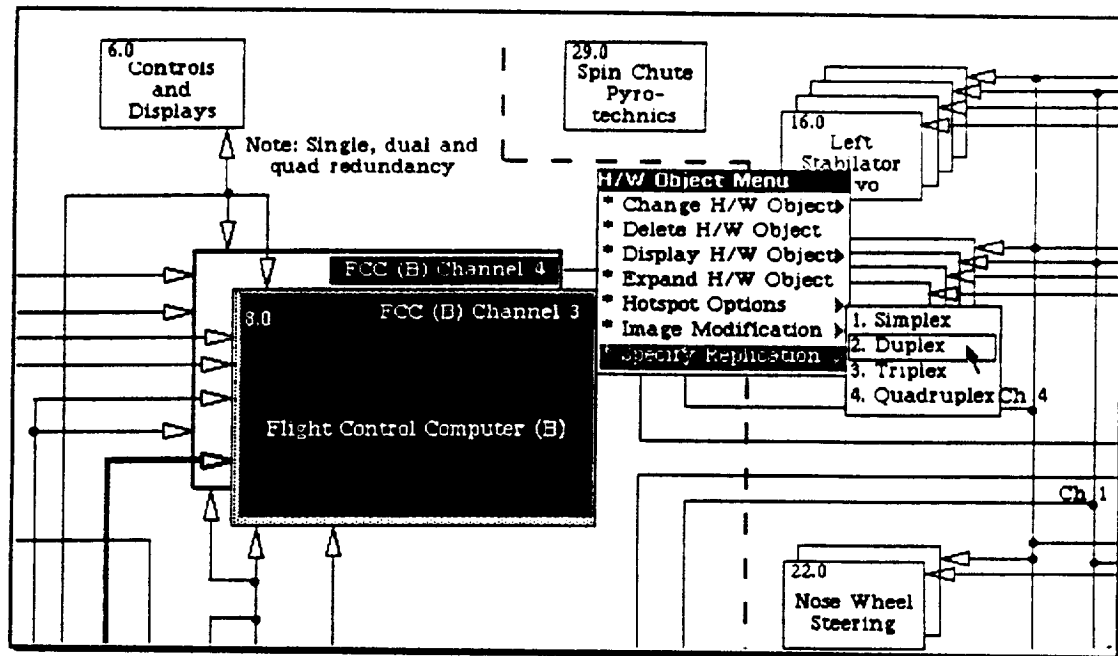


Figure 3.1.1-10. H/W Object Menu Replication Authoring Options

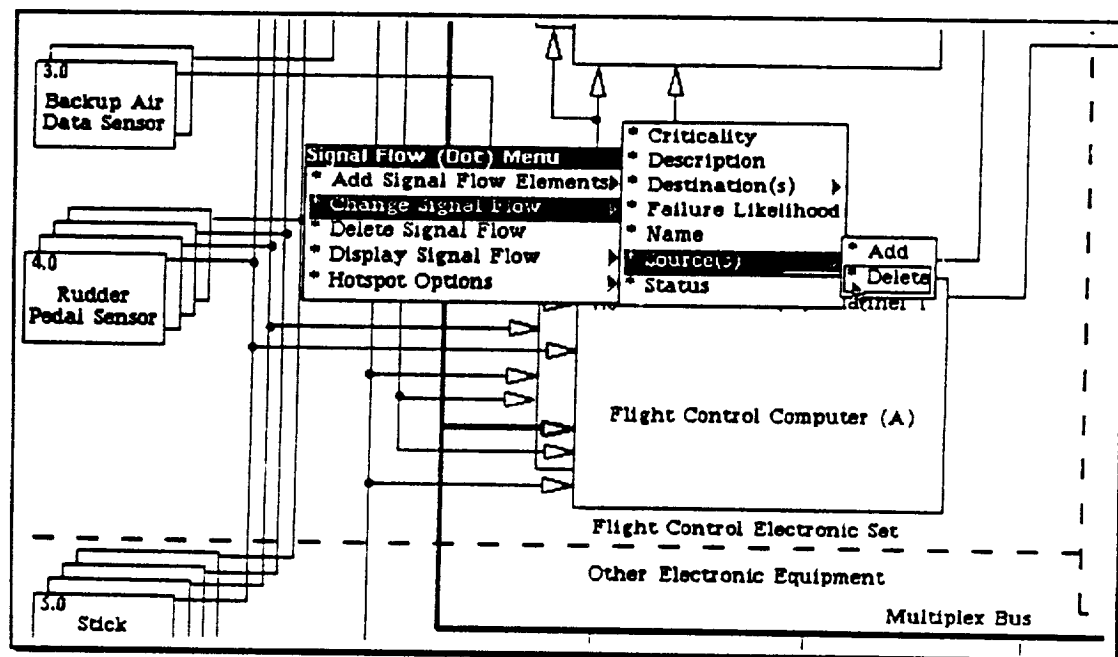


Figure 3.1.1-11. S/F Object Menu Change Authoring Options

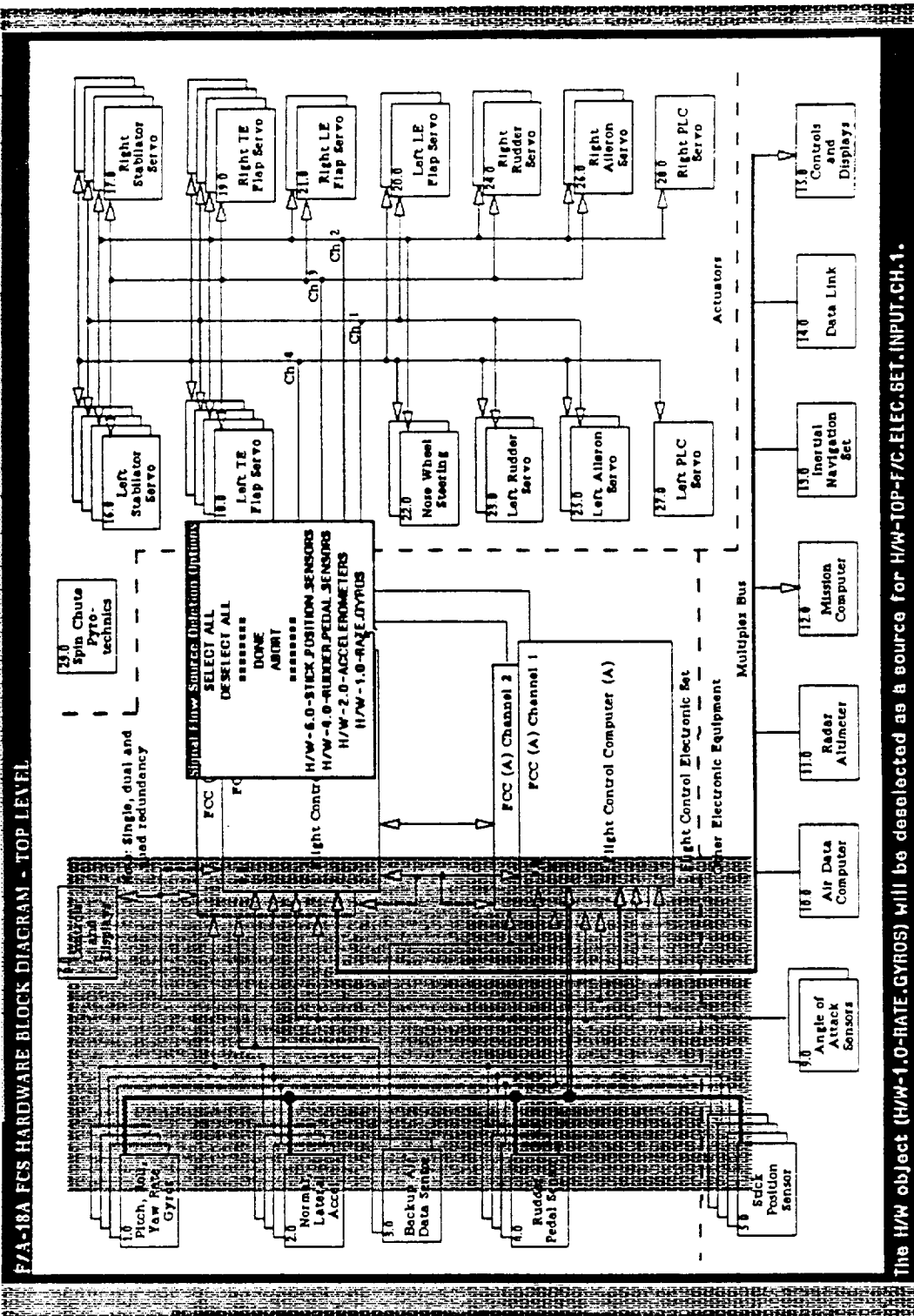


Figure 3.1.1-12. Typical S/F Object Source Deletion Pop Up Menu

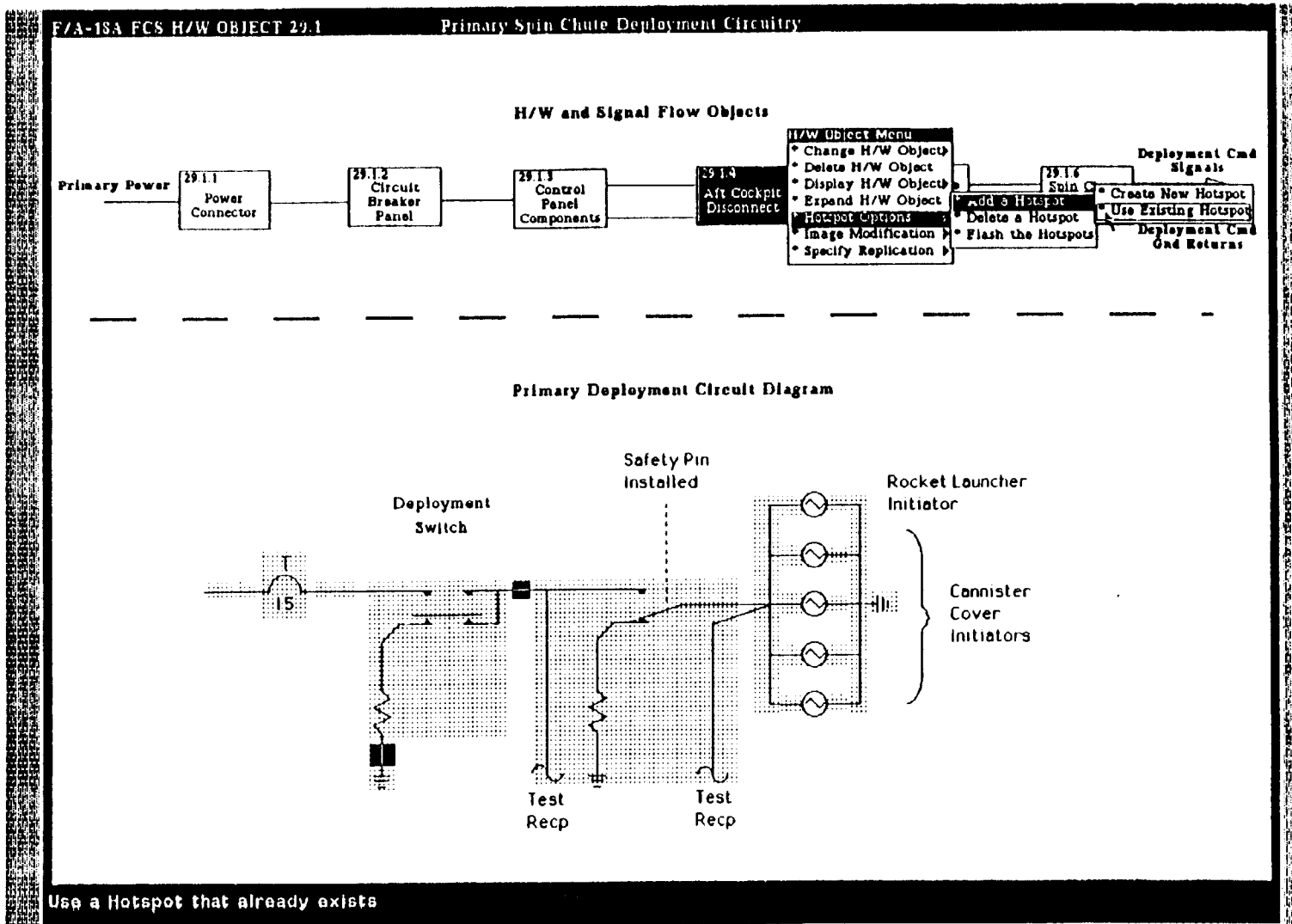


Figure 3.1.1-13. H/W Object Menu Hot Spot Authoring Options

### **3.1.2 Browsing Concepts**

Browsing methods have been created which are designed to support exploratory information seeking strategies. They are intended to support individual perspectives and work within a framework which supports a thinking process consistent with the cognitive theory of spreading activation for search. This approach may also be seen as an informal one which relies upon a form of serendipity to find the desired information. As the user navigates from object to object, content-oriented displays have been supplied to inform him/her.

The semantic network provides a roadmap which supports a search which is structured by the methodologies. It tends to highlight the semantic and physical relationships as well as the connectivity inherent to the problem domain. The structured menu-driven browsing mechanisms are rapid and incremental (a sense of being linear), however they also possess non-linear hypertext-like capabilities. These non-linear mechanisms allow the user to skip around within a realm and to switch to an entirely different realm.

A top level display is provided which serves as an introduction to the problem domain and clearly depicts the integration of the design knowledge associated with system design, H/W design, S/W design and utilities design. This display, depicted in Figure 3.1.2-1, gives the user access to any one of these four realms of knowledge.

#### 3.1.2.1 Structured Analysis Browsing Methods

##### **DFD Browsing Options**

Menu-driven browsing methods are provided for the DFD diagrams and for the individual objects. These DFD methods provide a global capability to browse the knowledge base in a non-linear fashion. The major options which are available for global browsing when a DFD is moused are indicated in Figure 3.1.2-2. These options allow the user to explore the connectivity of the objects, switch to another DFD, search a dictionary for an object definition, examine a hierarchy graphically or switch to another knowledge realm. In Figure 3.1.2-2, the *Connectivity* option has been selected, which gives the user the ability to explore source/destination connections for dataflow objects and input/output connections for process objects.

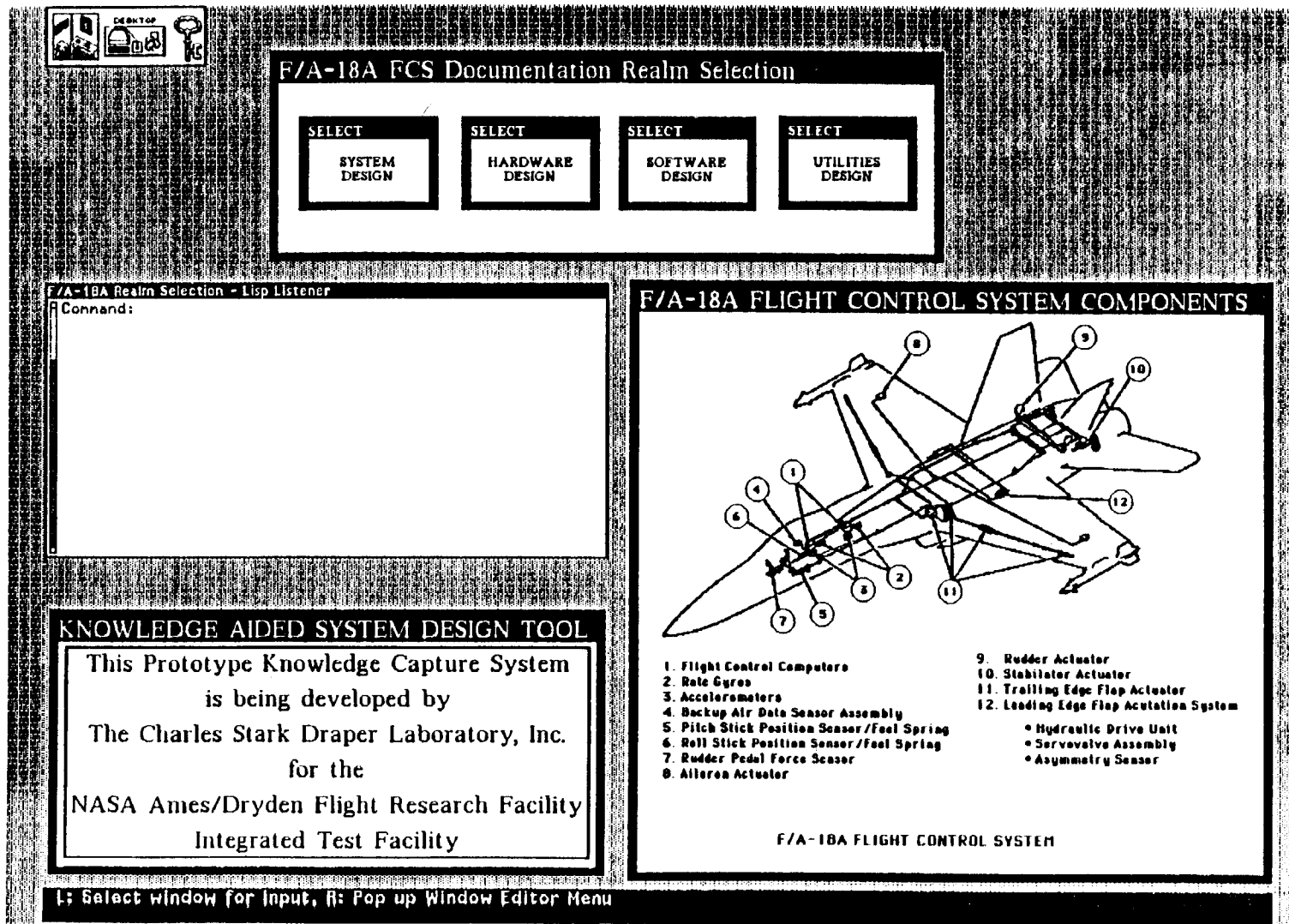


Figure 3.1.2-1. Top Level KCS Display Panel



## Knowledge Capture Concepts

The hierarchy and connectivity displays for processes are depicted in Figure 3.1.2-6. The hierarchy display graphically depicts the decomposition of processes within the system design realm. The process connectivity is depicted by a table of inputs and outputs on a process by process basis. Both of the displays are scrollable. The graphical hierarchy is mouse sensitive and allows the user to explore further the properties of a selected object. Figure 3.1.2-7 displays the hierarchy and connectivity for data flow objects in a similar format. In this case, only the hierarchy for the *actuator commands* data flow object is displayed so as to focus on the decomposition of a specific high level data flow object.

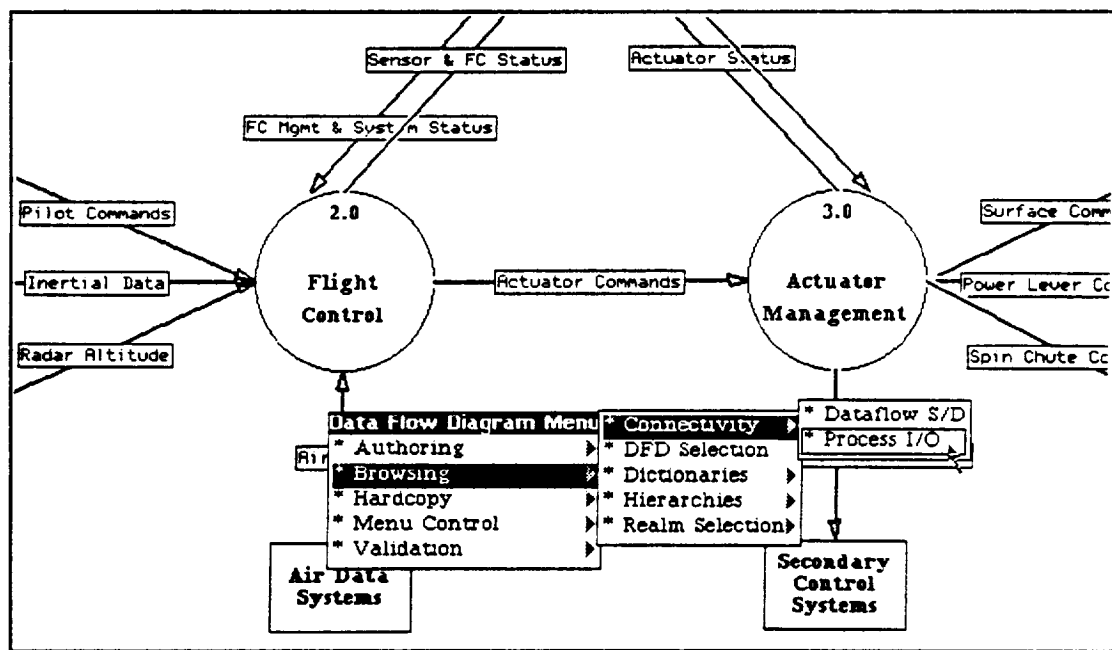


Figure 3.1.2-2. DFD Browsing Options

The DFD selection browsing option allows the user to select any one of the other DFDs within the realm. Figure 3.1.2-8 depicts the pop-up menu displayed when the *DFD Selection* option is moused. This menu allows the user to browse the DFDs and to select the one of interest.

Figure 3.1.2-9 depicts the *Dictionaries* options. Separate dictionaries are provided for data flow objects, external objects and process objects. Figure 3.1.2-10 depicts the pop-up menu displayed when the data flow dictionary is requested. It allows the user to browse

## Knowledge Capture Concepts

through the dictionary of data flow objects on a DFD by DFD basis. Subsequent selection of a particular data flow object will display a text description. Similar mechanisms are provided for the process and external dictionaries.

The objects defined within a realm are organized as structured hierarchies. The DFD menu *Hierarchies* browsing option is depicted in Figure 3.1.2-11. Figures 3.1.2-12 and 3.1.2-13 indicate the hierarchical decomposition displays provided for the user when the data flow hierarchy or the process hierarchy is selected. The objects depicted on the displays are mouse sensitive and can be moused to explore in greater depth. An adjustable window size and scroll bars allow the user the needed flexibility to explore these large hierarchies which are expected to consist of 100s of objects or more.

The *Realm Selection* menu item is depicted in Figure 3.1.2-14. It allows the user to select any one of the other realms (H/W Design, S/W Design or Utilities Design).

### **Process Object Browsing Options**

The process object menu provides a capability to browse local process knowledge as opposed to the global scale of knowledge access provided by the DFD menus. The major

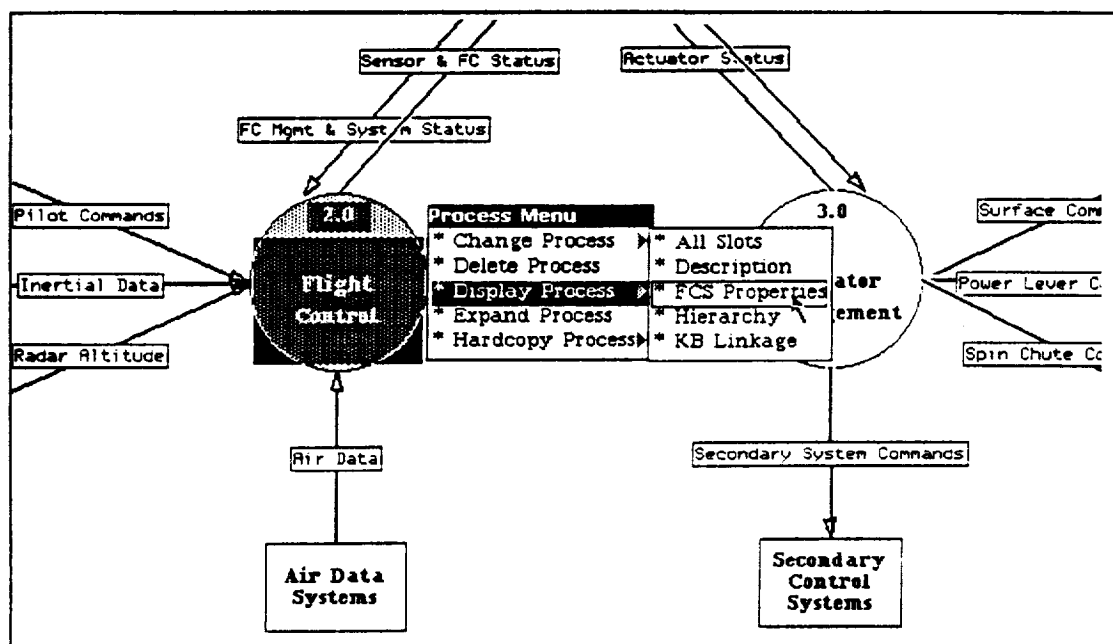


Figure 3.1.2-3. Process Object Browsing Options

## Knowledge Capture Concepts

options which are available for the local examination of a process object when the process icon is moused are indicated in Figure 3.1.2-3. These options allow the user to either explore all of the properties of the selected process or else to be more selective and focus on a more specific set of properties.

Figure 3.1.2-15 depicts the display when all of the properties are selected for viewing. A scrolling capability is provided in this display which allows the user to explore the many aspects of the selected process object. In addition, to the listing of the properties a focused display of the process within the semantic network is provided. This display depicts the position of the process in the process object hierarchy with many members of the process hierarchy suppressed so as to focus directly on the decomposition of the selected process. This hierarchy display depicts only the direct line ancestors and all the descendents for the selected process object.

The more selective browsing options on the process menu allow the user to be more specific when examining the process properties. These options allow the user to explore restricted sets of properties, such as the *FCS Properties* (Figure 3.1.2-16 ) and the *KB Linkage* (Figure 3.1.2-17). Thus the user is able to examine a specific subset of the process properties relevant to a narrower interest within the problem domain. Here again, as with the "all properties" display, a focused hierarchy of the process within the semantic network is provided.

In addition, it is possible to selectively display a text description (Figure 3.1.2-18), a focused hierarchy and the decomposition (expansion) of the object into a DFD. The focused hierarchy is the same graphical display supplied when a properties option is selected as depicted in Figures 3.1.2-15 through 3.1.2-17. The expansion display option descends one level in the DFD hierarchy and the decomposition of the process object is displayed, as illustrated in Figure 3.1.2-19.

### **Data Flow Object Browsing Options**

The major options which are available for the local examination of a data flow object when a data flow icon is moused are indicated in Figure 3.1.2-4 . These options are nearly identical to those described for process objects. Here again, the browsing options allow the user to explore all of the properties of the selected data flow object or to be more selective.

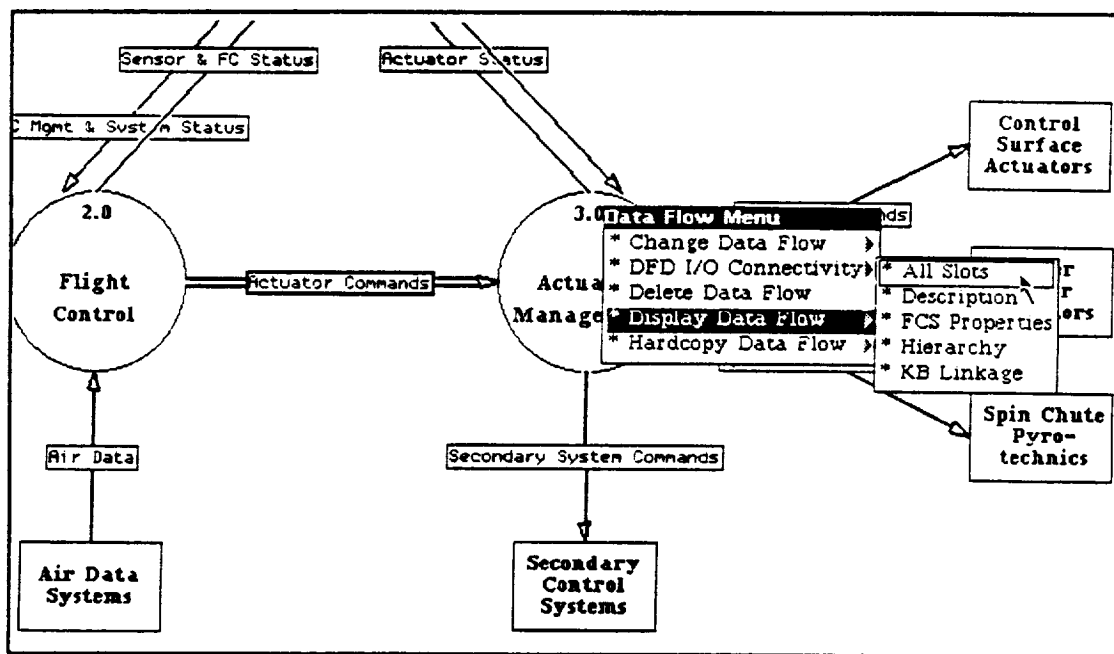


Figure 3.1.2-4. Data Flow Object Browsing Options

Figure 3.1.2-20 depicts the display when all of the properties are selected for viewing. This is the only data flow object browsing display described in detail here because of the purposeful similarity of the data flow browsing mechanisms with the process object browsing mechanisms, which were previously discussed with greater detail. Note the similarity of this display with that shown for process objects given in Figure 3.1.2-15. Only object specific properties and the nature of the focused data flow hierarchy have changed. For example, data flow objects possess source/destination characteristics, not inputs/outs characteristics. The focused hierarchy provides an interesting view of the unsymmetrical decomposition of data flow objects which deserves further comment.

This focused hierarchy display is similar to the display mechanized for process objects display in that it also looks at a restricted portion of the semantic network and depicts only the direct line ancestors and all the descendents for the selected object. In addition, it is possible to see how the source data and destination data for the selected data flow object are decomposed (split) within the DFD hierarchy. However, the decomposition is not necessarily symmetric even though the source data and the destination data for a given data

## Knowledge Capture Concepts

flow object must be one and the same. The typical decomposition depicted in Figure 3.1.2-20 clearly depicts an unsymmetrical (and not atypical) data flow object decomposition. The decomposition of the actuator commands data flow object located in the top level DFD is shown to result in only a single source data flow object located in DFD-2.0 and three destination data flow objects in DFD-3.0. The focused hierarchical display, which shows the further decomposition in lower level DFDs, allows the browser to view this unsymmetrical decomposition as it continues to occur in lower levels.

### **External Object Browsing Options**

The major options which are available for the local examination of an external object when an external object icon is moused are indicated in Figure 3.1.2-5. These options are nearly identical to those previously described for process objects (See Figure 3.1.2-3) and data flow objects (See Figure 3.1.2-4). Here again, the browsing options allow the user to explore all of the properties of the selected external object or to be more selective.

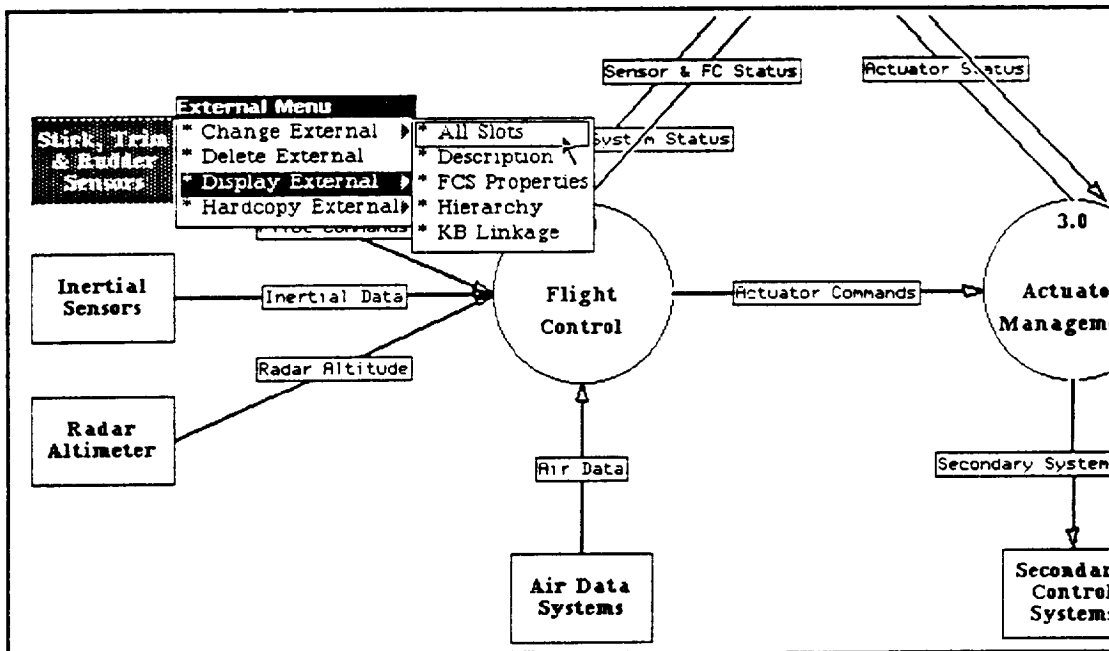


Figure 3.1.2-5. External Object Browsing Options

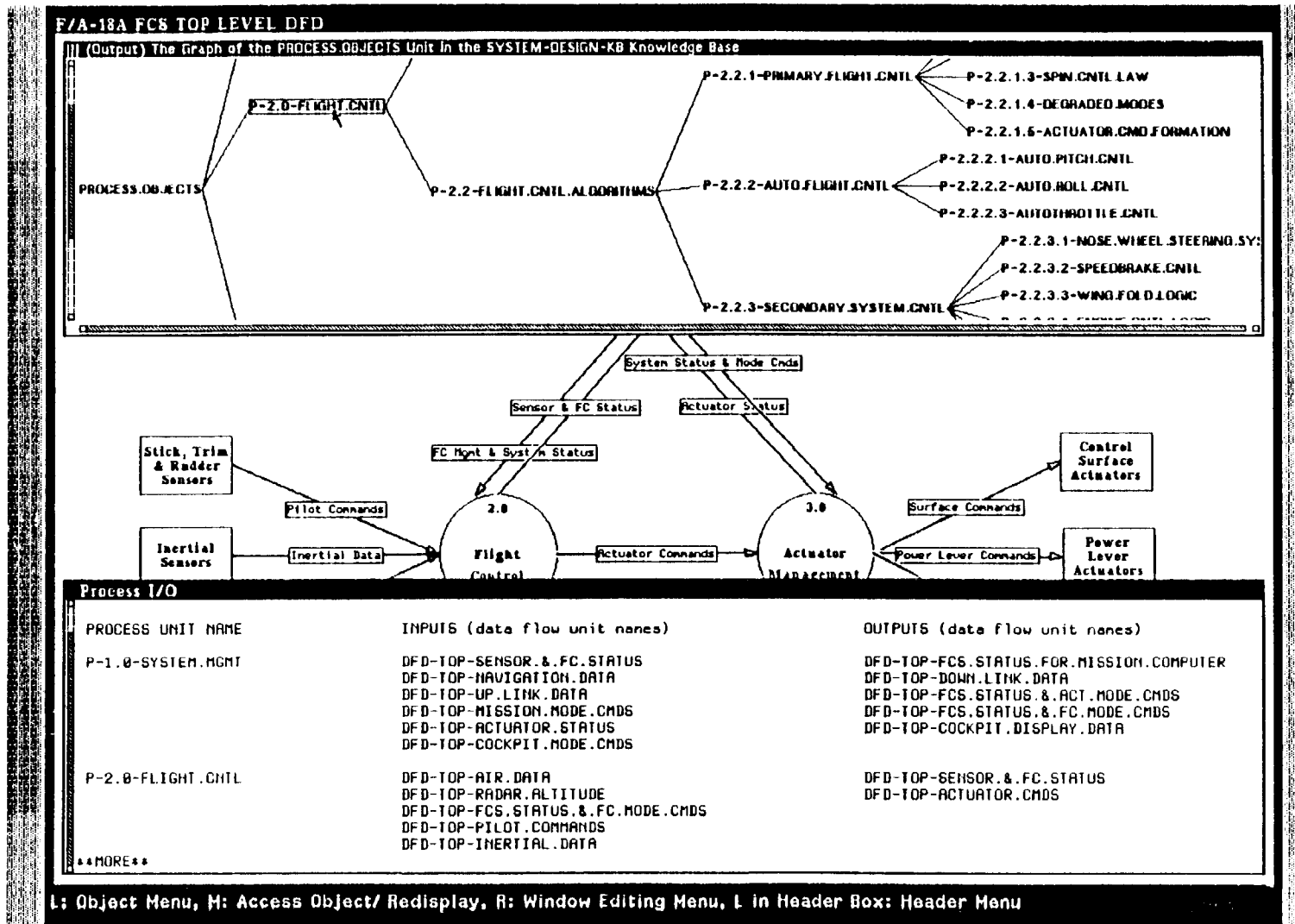


Figure 3.1.2-6. Process Hierarchy and Connectivity Displays

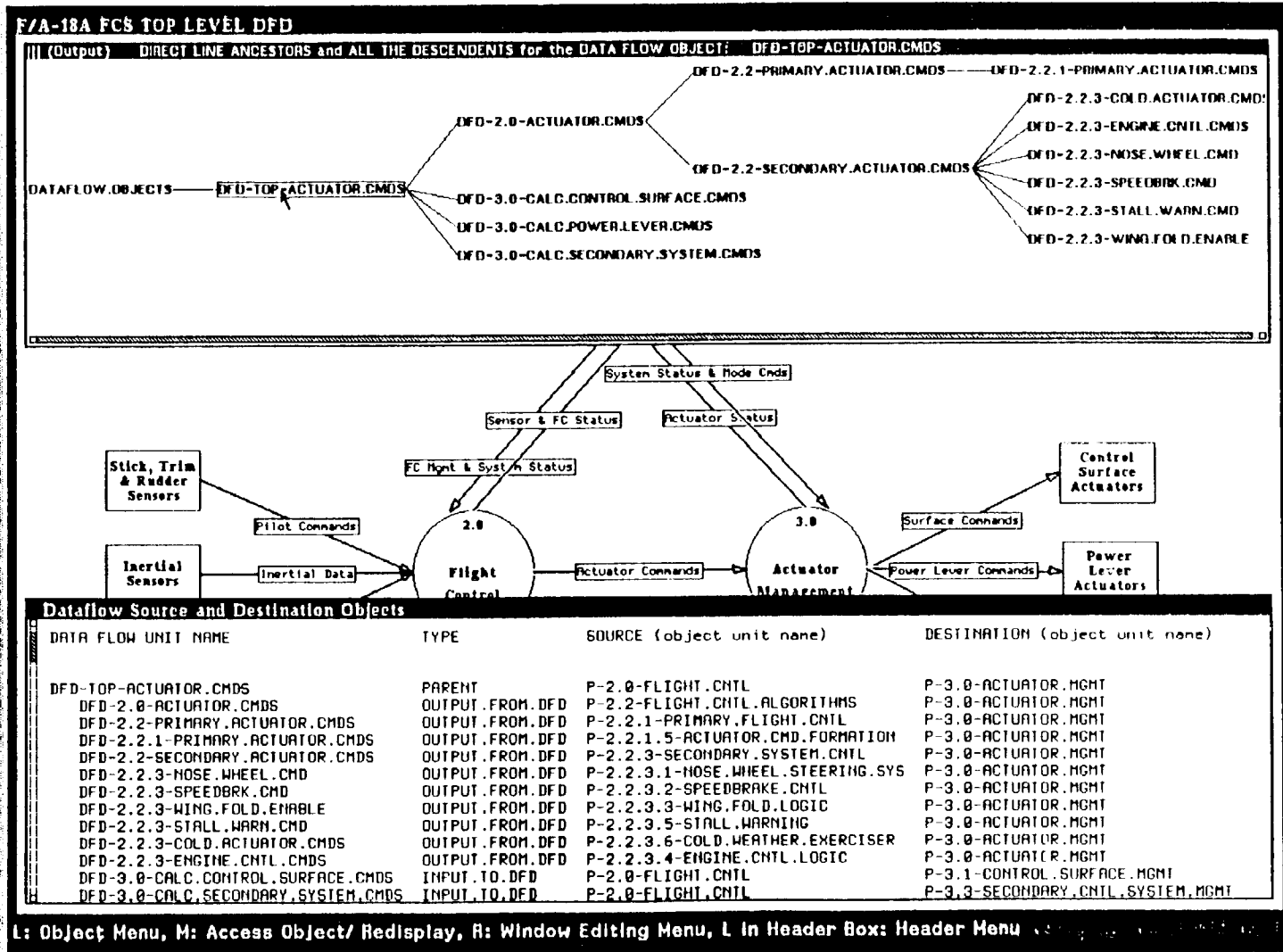


Figure 3.1.2-7. Data Flow Hierarchy and Connectivity Displays

## Knowledge Capture Concepts

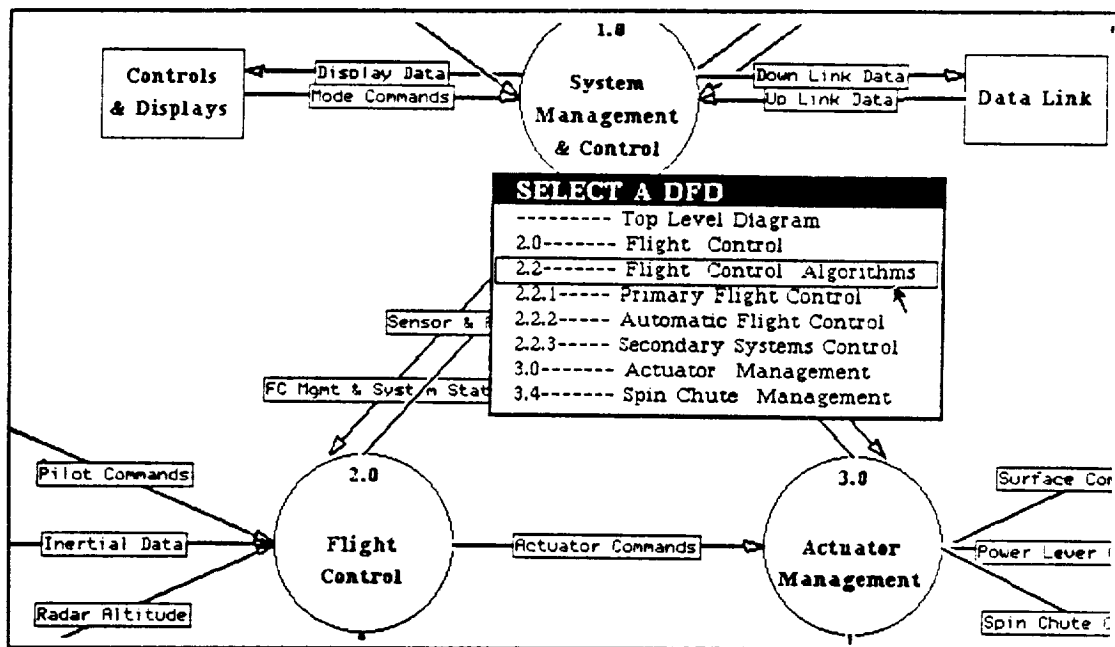


Figure 3.1.2-8. DFD Selection Menu

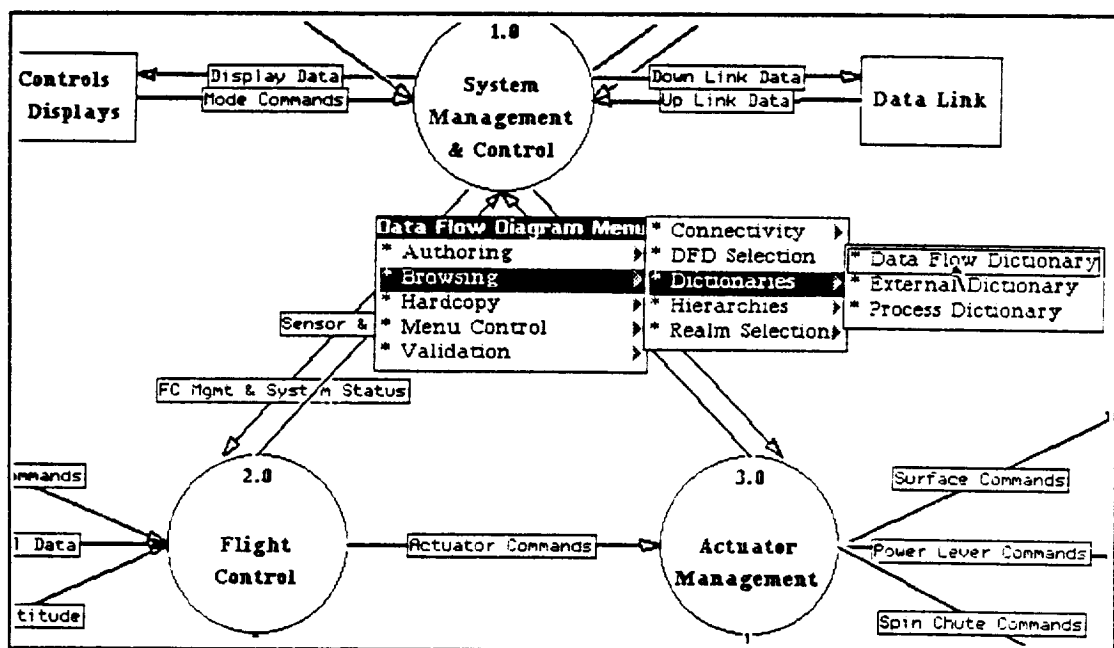


Figure 3.1.2-9. Dictionary Browsing Options



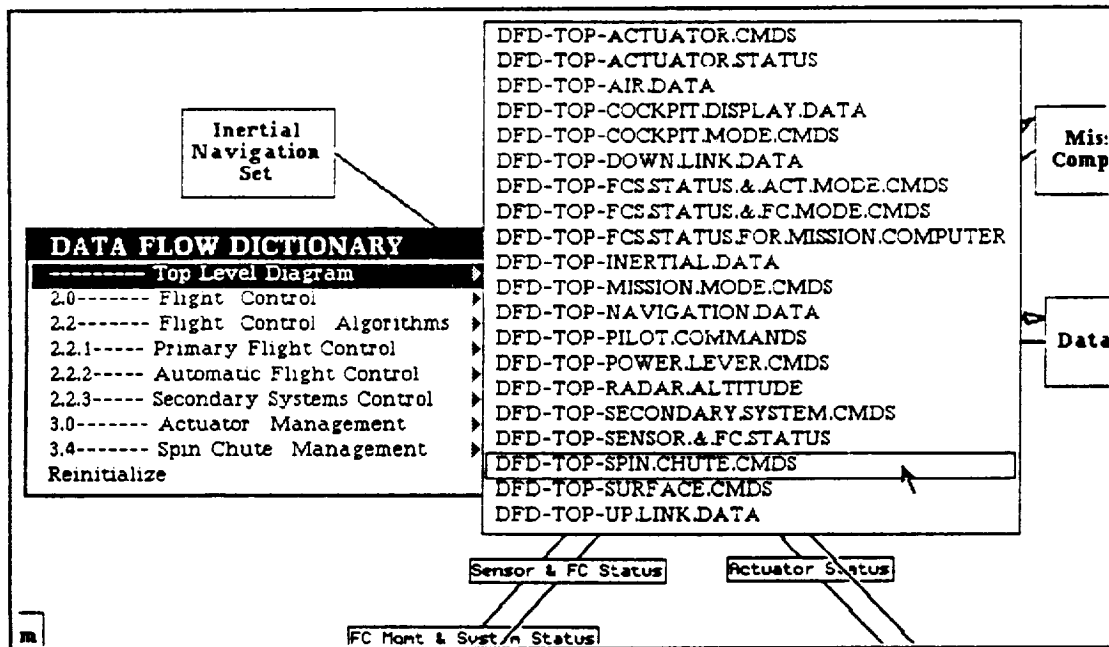


Figure 3.1.2-10. A Typical Data Flow Dictionary Menu

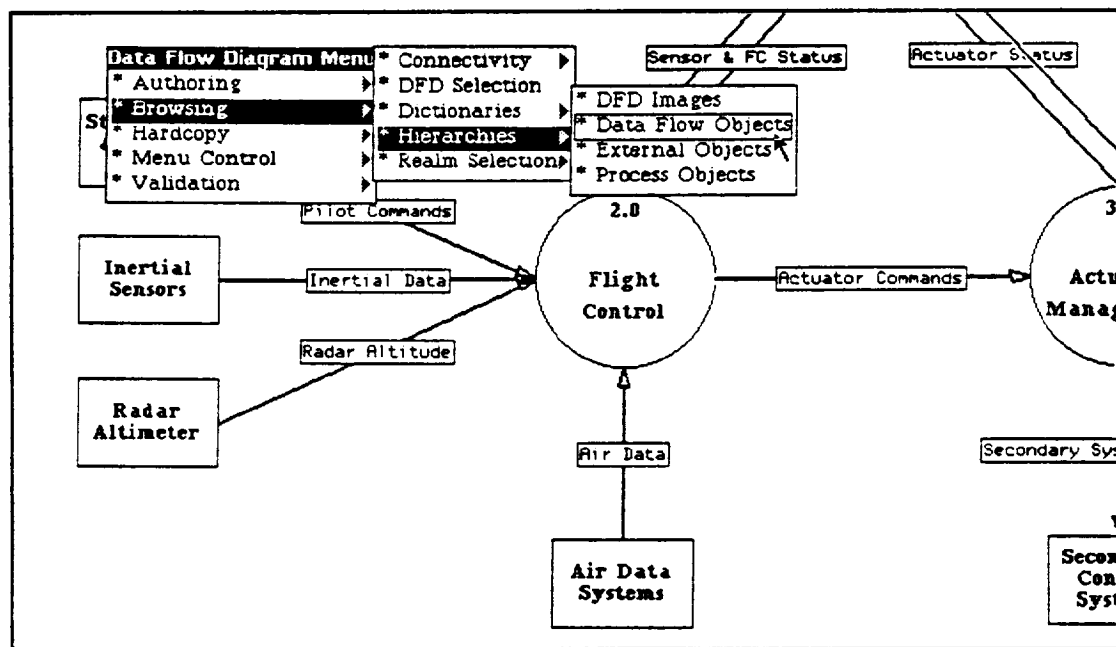


Figure 3.1.2-11. Hierarchy Browsing Options



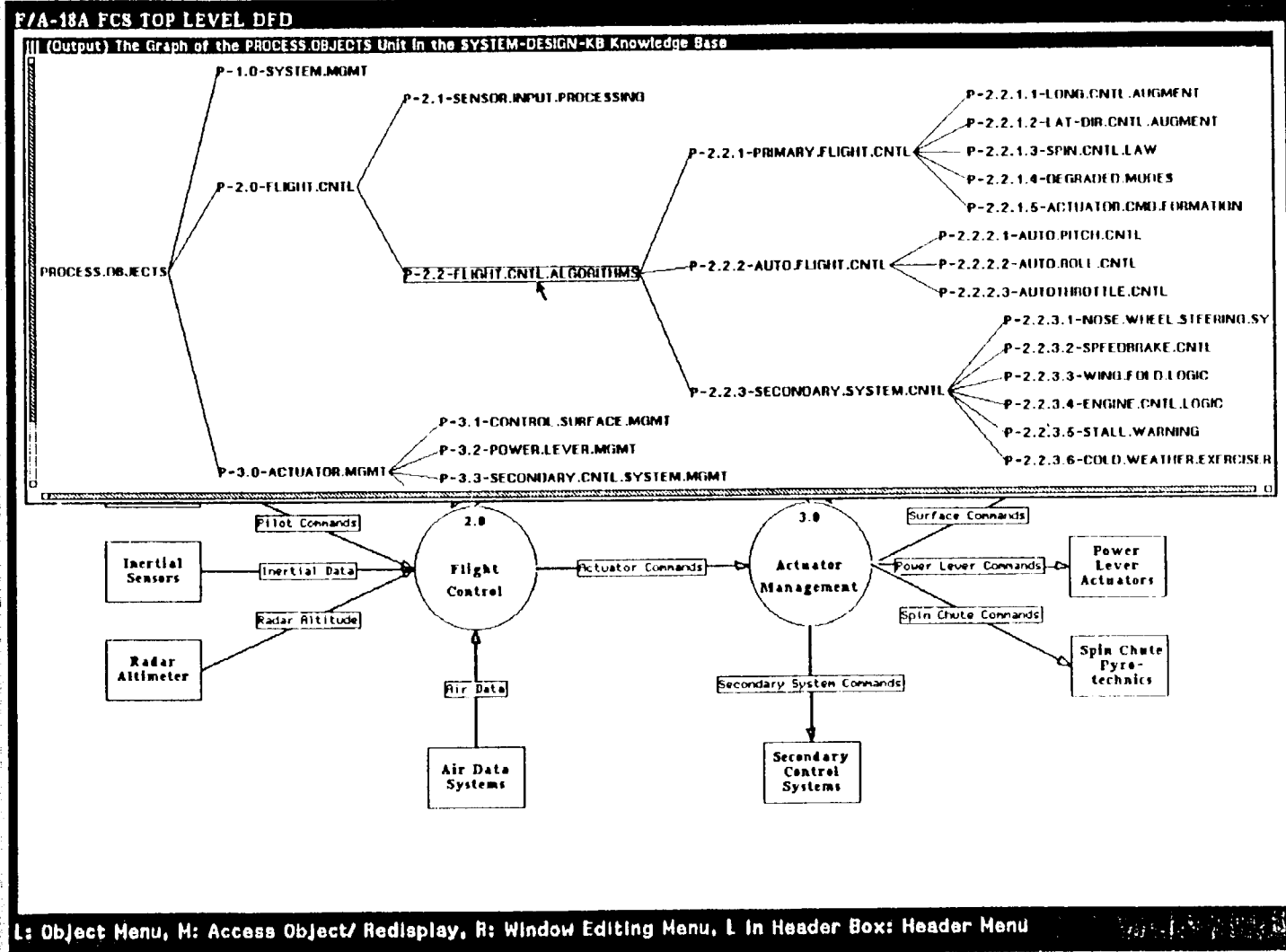


Figure 3.1.2-13. Process Hierarchy (Decomposition) Display

# Knowledge Capture Concepts

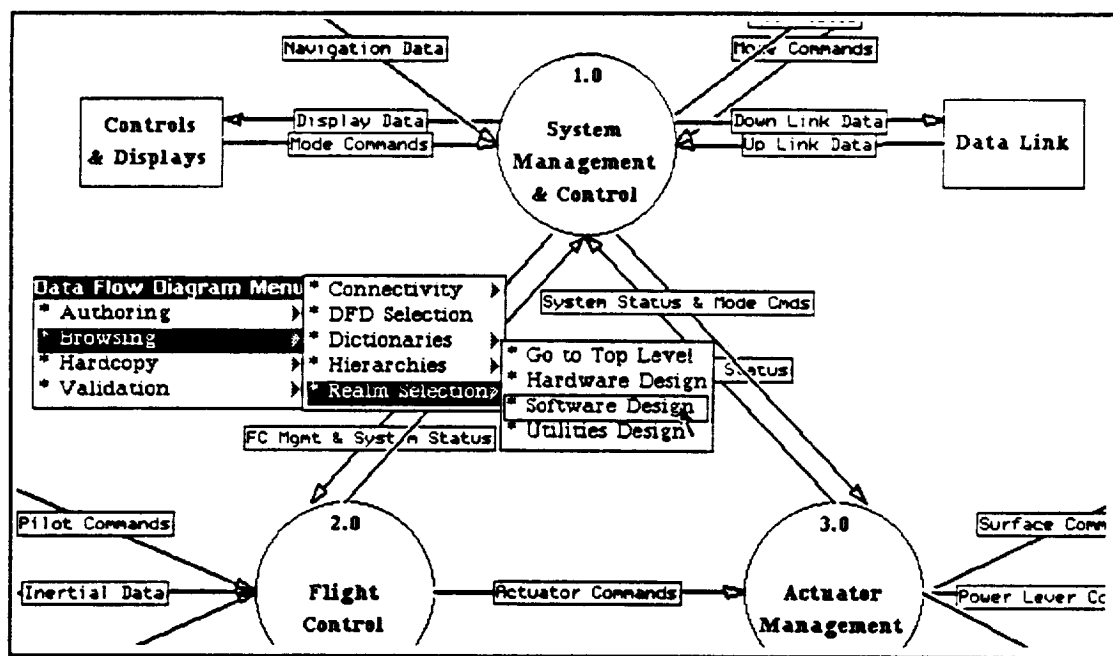


Figure 3.1.2-14. Realm Selection Browsing Options

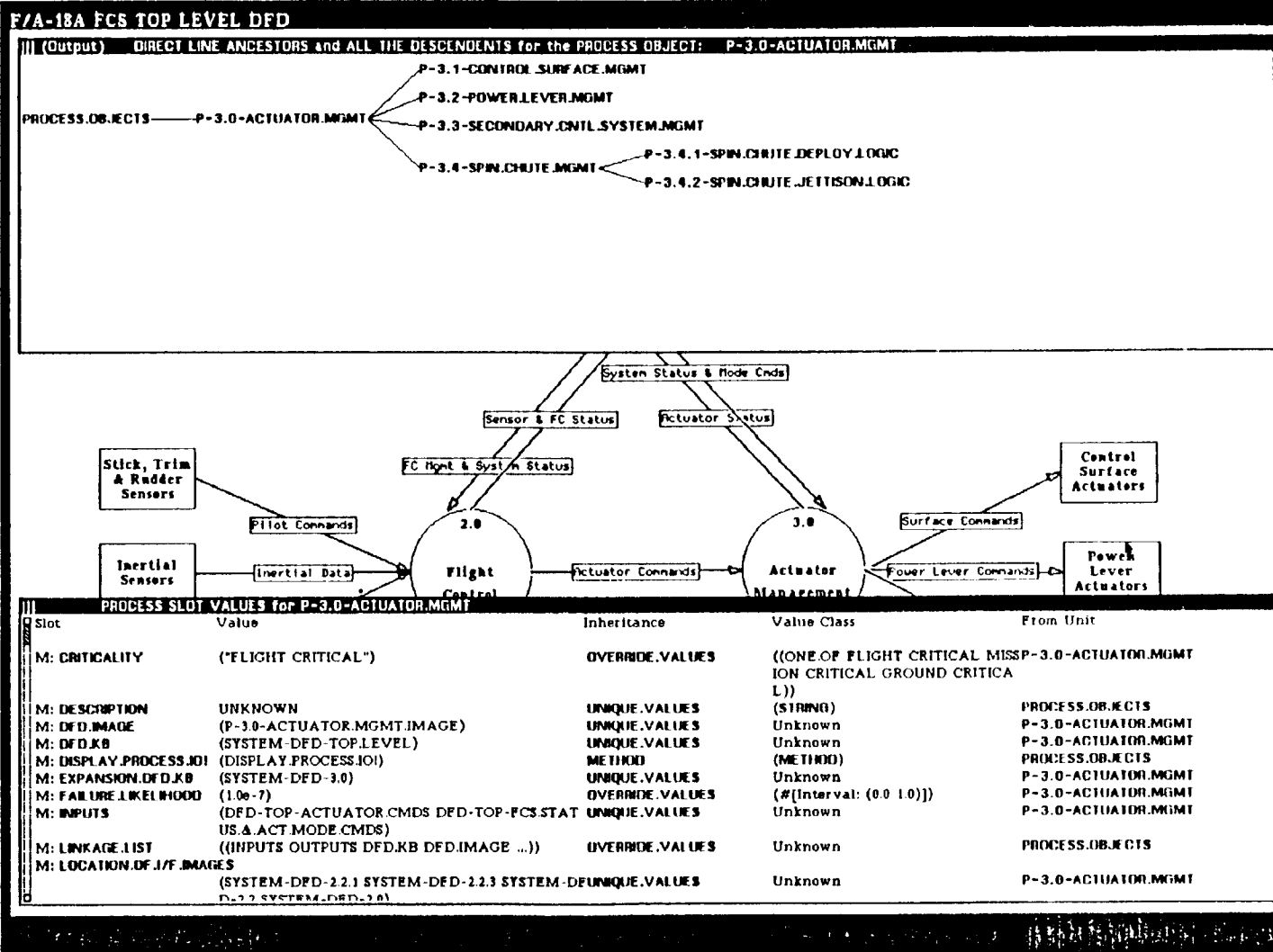


Figure 3.1.2-15. An All Properties Display for a Process Object

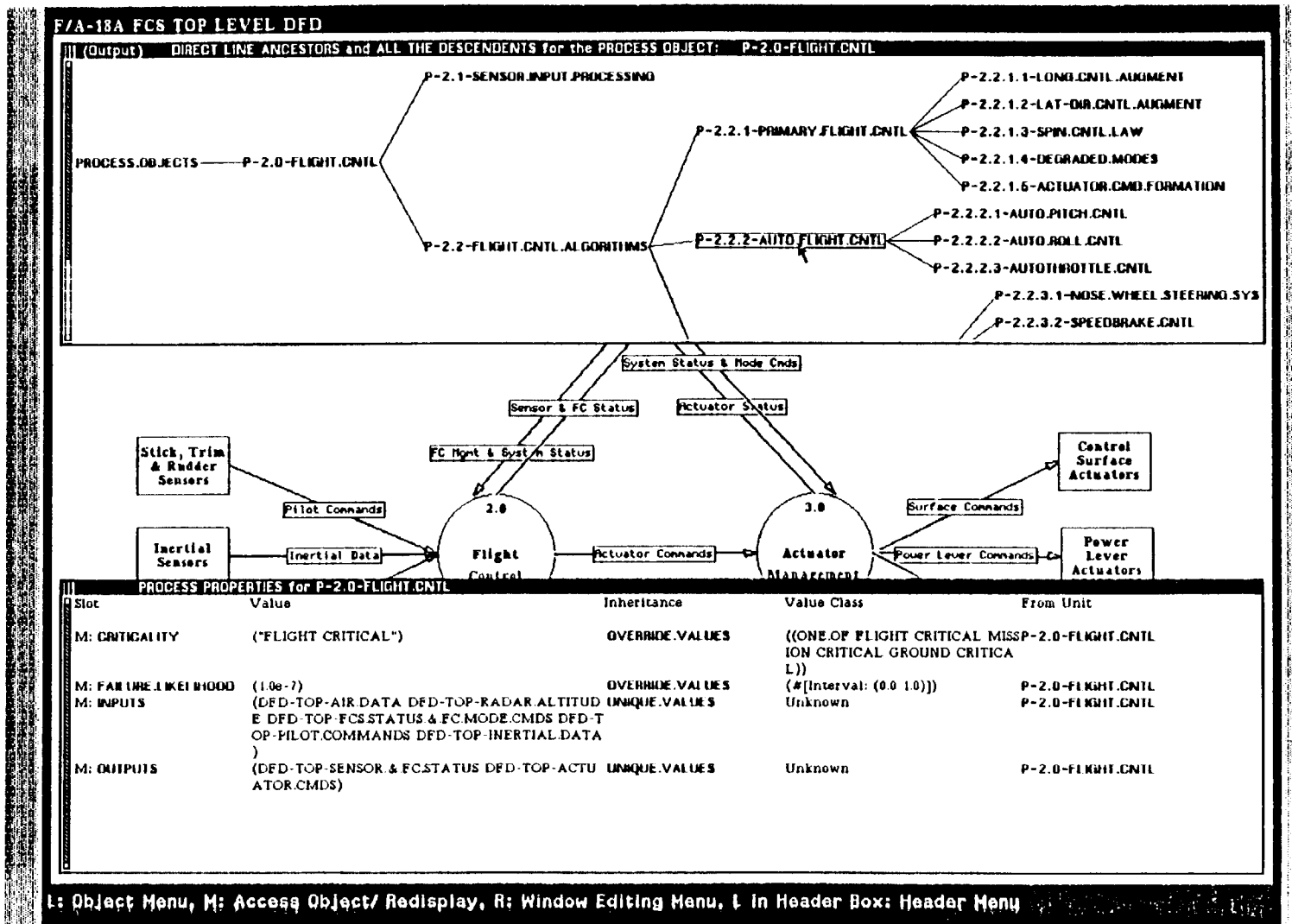


Figure 3.1.2-16. An FCS Properties Display for a Process Object

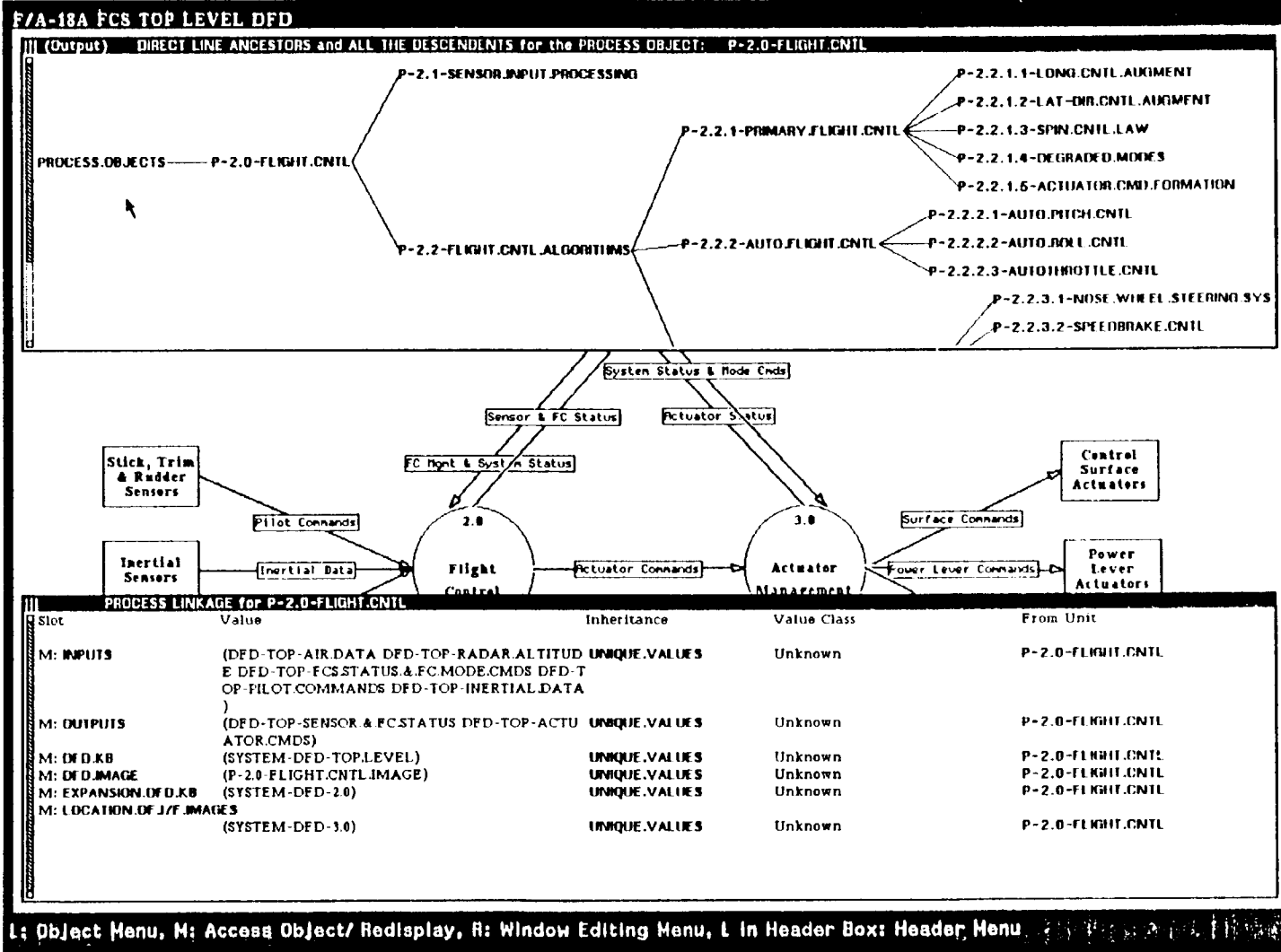
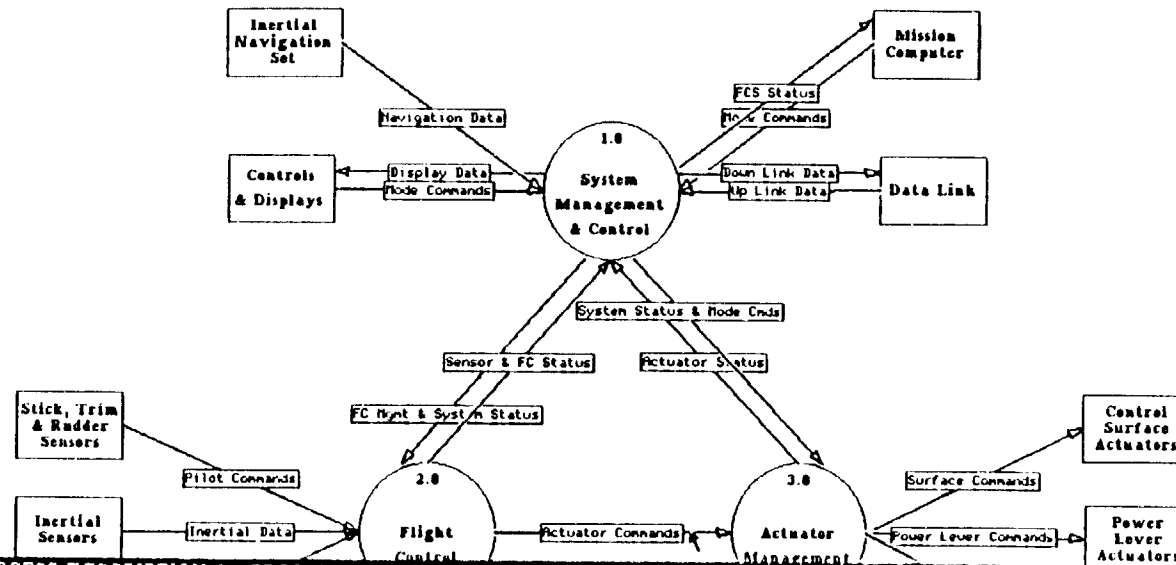


Figure 3.1.2-17. A KB Linkage Properties Display for a Process Object

F/A-18A FCS TOP LEVEL DFD



PROCESS DESCRIPTION for PROCESS P-2.0-FLIGHT.CNTL

The F-18 primary flight control system is a control augmentation system configuration which is implemented using fly-by-wire techniques. All control law computations are performed by four digital computers working in parallel. The digital computers are used in conjunction with redundant electrohydraulic servoactuators and analog sensors to provide two fail operate primary control capability. Backup mechanical control of the stabilator surfaces is available in the event of three digital processor failures or total electrical failure. Backup open loop analog control of the aileron and rudder surfaces is also available if the digital processors fail.

The control augmentation system uses gain scheduling, crossaxis interconnects (e.g., rolling survice to rudder) and closed loop control of aircraft response to enhance flying qualities and augment basic airframe stability. Angle of attack and air data parameters are used for gain scheduling the control system to accommodate varying flight conditions. Fixed gain values provide safe control upon failure of angle of attack or air data sensing. Out of control (spin) is automatically sensed and the control laws are reconfigured to facilitate recovery.

Digital direct electrical link control laws provide open loop control if the motion feedback sensors fail. The direct electric

Figure 3.1.2-18. A Text Description Display for a Process Object



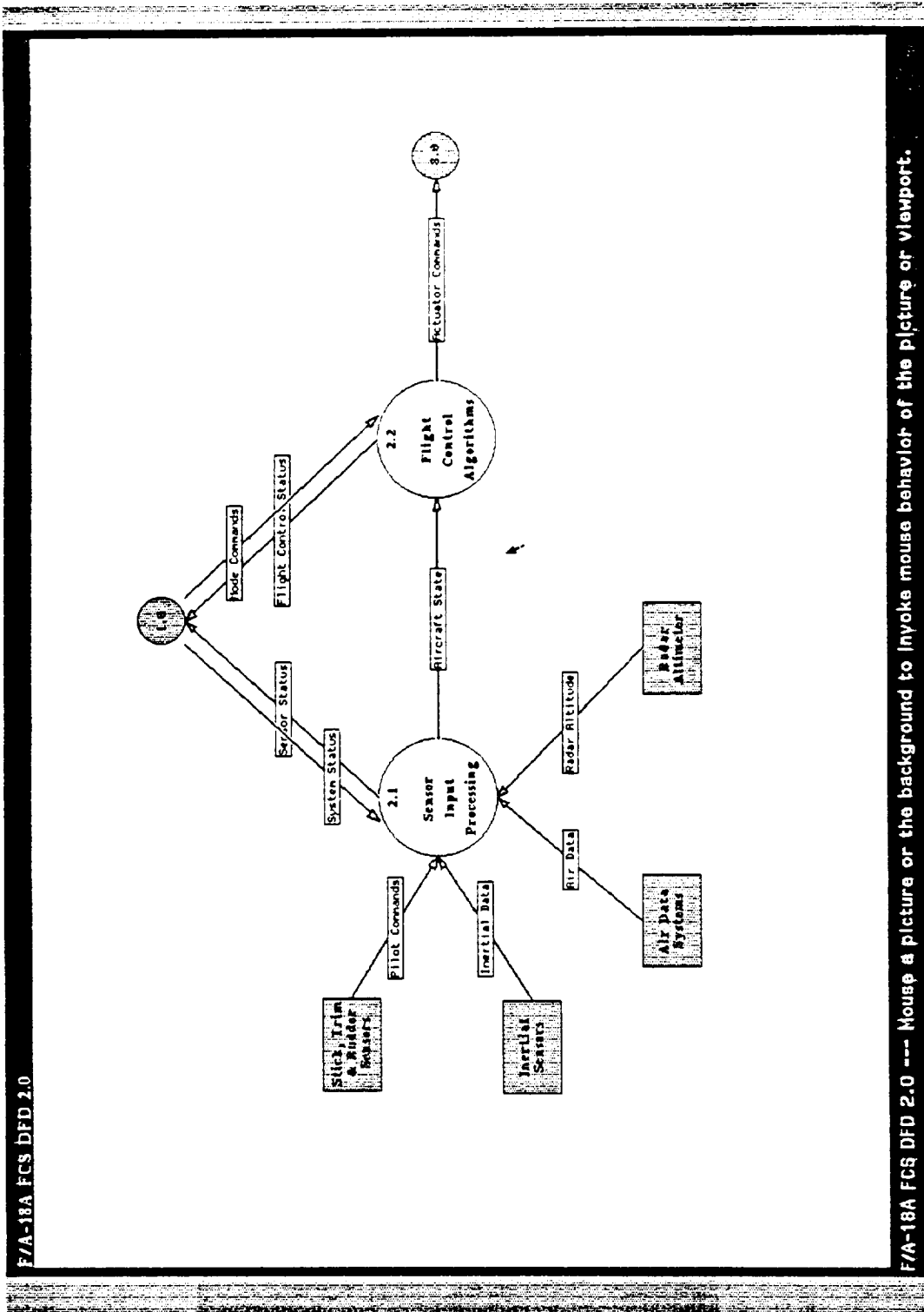


Figure 3.1.2-19. An Expansion Display for a Process Object

1. Object Menu, M: Access Object/Redisplay, R: Window Editing Menu, I in Header Box: Header Menu



### 3.1.2.2 H/W Design Browsing Methods

#### H/W Diagram Browsing Options

Menu-driven browsing methods are provided for the H/W diagrams and for the individual objects. The H/W diagram methods provide a global capability to browse the knowledge base in a non-linear fashion. The major options, which are available for global browsing when a H/W diagram is moused, are indicated in Figure 3.1.2-21. These options allow the user to switch to another H/W diagram, examine a hierarchy graphically, explore the behavioral models or switch to another knowledge realm.

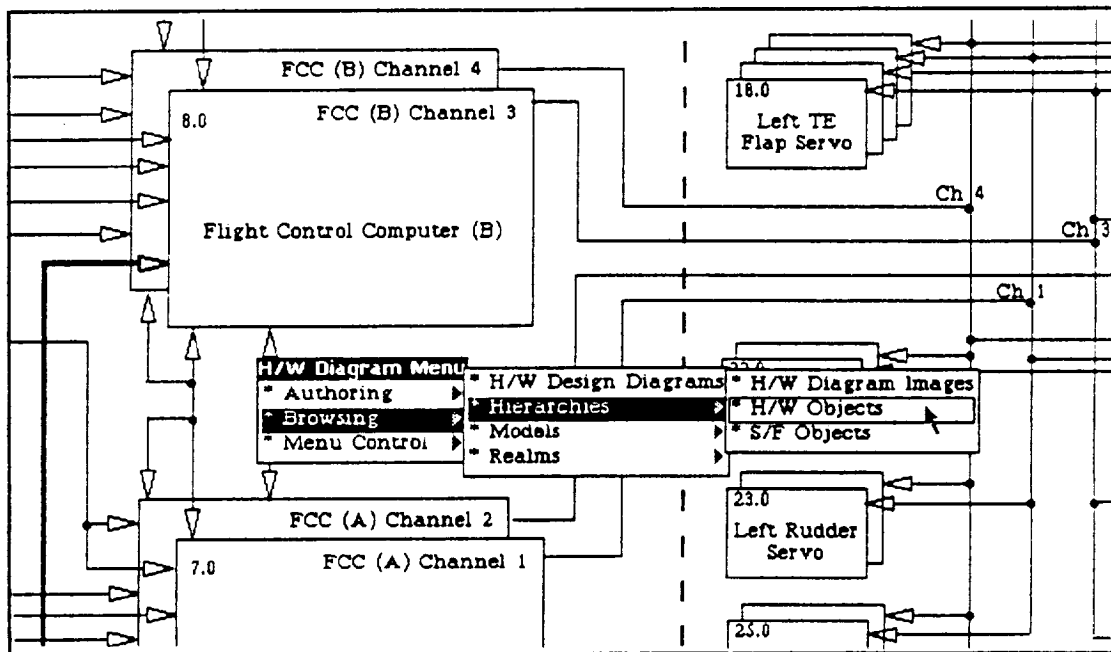


Figure 3.1.2-21. H/W Diagram Browsing Options

The *H/W Design Diagrams* browsing option allows the user to select any one of the other H/W diagrams within the realm. When this option is moused, a pop-up menu of choices is displayed as depicted in Figure 3.1.2-24. This menu allows the user to browse the H/W diagrams and to select the one of interest. The *Realms* browsing option is depicted in Figure 3.1.2-25. It allows the user to select any one of the other realms (System Design, S/W Design or Utilities Design).

## Knowledge Capture Concepts

The objects defined within a realm are organized as structured hierarchies. The H/W diagram menu browsing option for these hierarchies is depicted in Figure 3.1.2-21. Figures 3.1.2-26 and 3.1.2-27 indicate the hierarchical decomposition displays provided for the user when the H/W objects hierarchy or the signal flow objects hierarchy is selected. The objects depicted on the displays are mouse sensitive and can be moused to explore in greater depth. An adjustable window size and scroll bars allow the user the needed flexibility to explore these large hierarchies which are expected to consist of 100s of objects or more.

The *Models* browsing option allows the user to request the display of models such as those described in Section 8. These models allow the user to explore dynamic models which indicate the functionality of the H/W and S/F objects.

### H/W Object Browsing Options

The H/W object menu provides a capability to browse local H/W object knowledge as opposed to the global scale of knowledge access provided by the H/W diagram menu. The major options which are available for the local examination of a H/W object when a

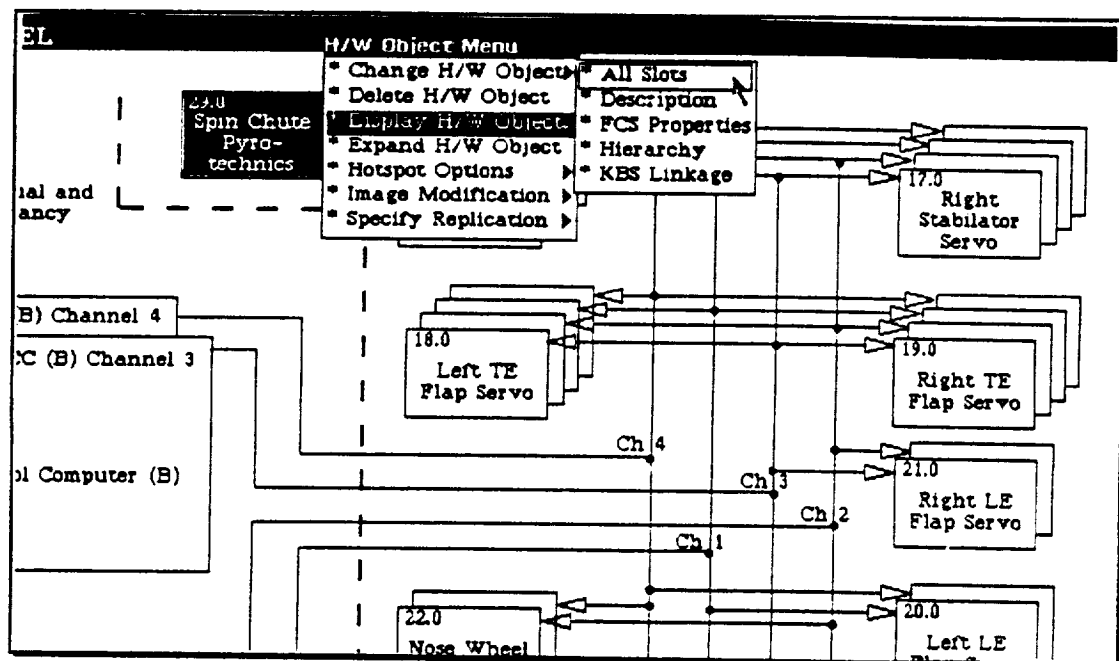


Figure 3.1.2-22. H/W Object Browsing Options

## Knowledge Capture Concepts

H/W object icon is moused are indicated in Figure 3.1.2-22. These options allow the user to either explore all of the properties of the selected H/W object or else to be more selective and focus on a more specific set of properties.

Figure 3.1.2-28 depicts the display when all of the properties are selected for viewing. A scrolling capability is provided in this display which allows the user to explore the many aspects of the selected H/W object. In addition, to the listing of the properties a focused display of the H/W object within the semantic network is provided. This display depicts the position of the H/W object in the H/W object hierarchy with many members of the H/W object hierarchy suppressed so as to focus directly on the decomposition of the selected H/W object. This hierarchy display depicts only the direct line ancestors and all the descendents for the selected H/W object.

The more selective browsing options on the H/W object menu allow the user to be more specific when examining the H/W object properties. These options allow the user to explore restricted sets of properties, such as the *FCS Properties* (Figure 3.1.2-29) and the *KBS Linkage* (Figure 3.1.2-30). Thus, the user is able to examine a specific subset of the properties relevant to a narrower interest within the problem domain. Here again, as with the *All Properties* display, a focused hierarchy of the H/W object within the semantic network is provided.

In addition, it is possible to selectively display a text description, a focused hierarchy and the decomposition (expansion) of the object into a H/W diagram. The *Hierarchy* display option activates the same graphical display supplied when a H/W property display option is selected as depicted in Figures 3.1.2-28 through 3.1.2-30. The *Expand H/W Object* option descends one level in the H/W diagram hierarchy and the decomposition of the H/W object is displayed, as illustrated by the typical level 2 H/W diagram depicted in Figure 3.1.2-31,

The *Hotspot Options* allow the user to explore the linkage between the selected object and an engineering drawing displayed on the same H/W diagram. A hotspot browsing option allows the user to flash the selected object and its related hotspots on the drawing as indicated in Figure 3.1.2-32.

## Knowledge Capture Concepts

### Signal Flow Object Browsing Options

The major options which are available for the local examination of a signal flow object when a signal flow icon is moused are indicated in Figure 3.1.2-23. These options are nearly identical to those described for H/W objects. Here again, the browsing options allow the user to explore all of the properties of the selected signal flow object or to be more selective.

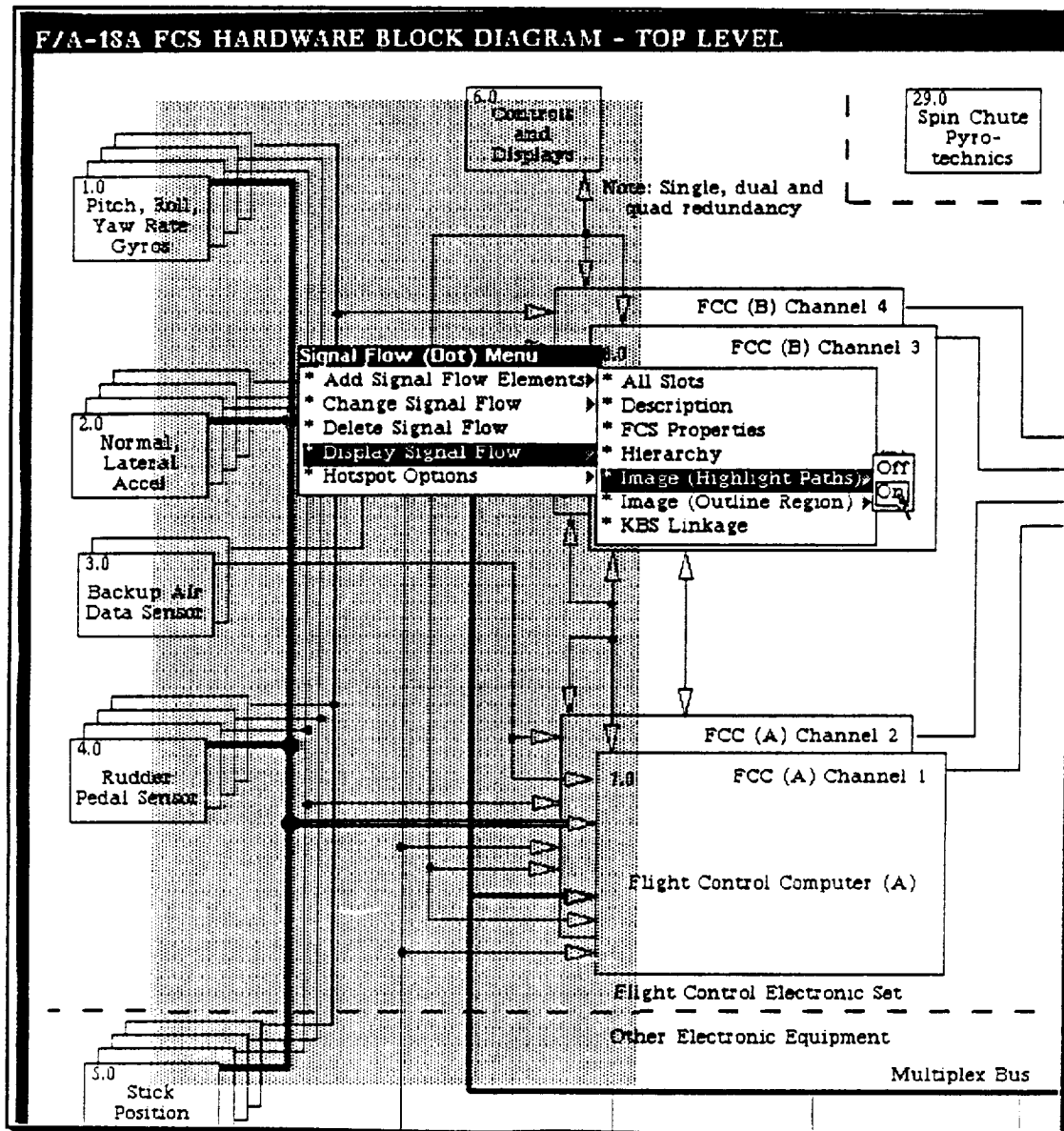


Figure 3.1.2-23. Signal Flow Object Browsing Options

### Knowledge Capture Concepts

Figure 3.1.2-33 depicts the display when all of the properties are selected for viewing. Note the purposeful similarity of the signal flow browsing mechanisms with the H/W object browsing mechanisms, given in Figure 3.1.2-22. Only object specific properties and the nature of the focused signal flow hierarchy have changed.

The signal flow objects are allowed to possess multiple inputs and outputs. The browsing methods allow the user to highlight and to outline the region of a signal flow. Figure 3.1.2-23 depicts a display in which the channel 1 inputs to a computer have been highlighted so as to clearly display a signal flow, which in this case has multiple inputs. The menu command used to obtain this display enhancement is also depicted in Figure 3.1.2-23.

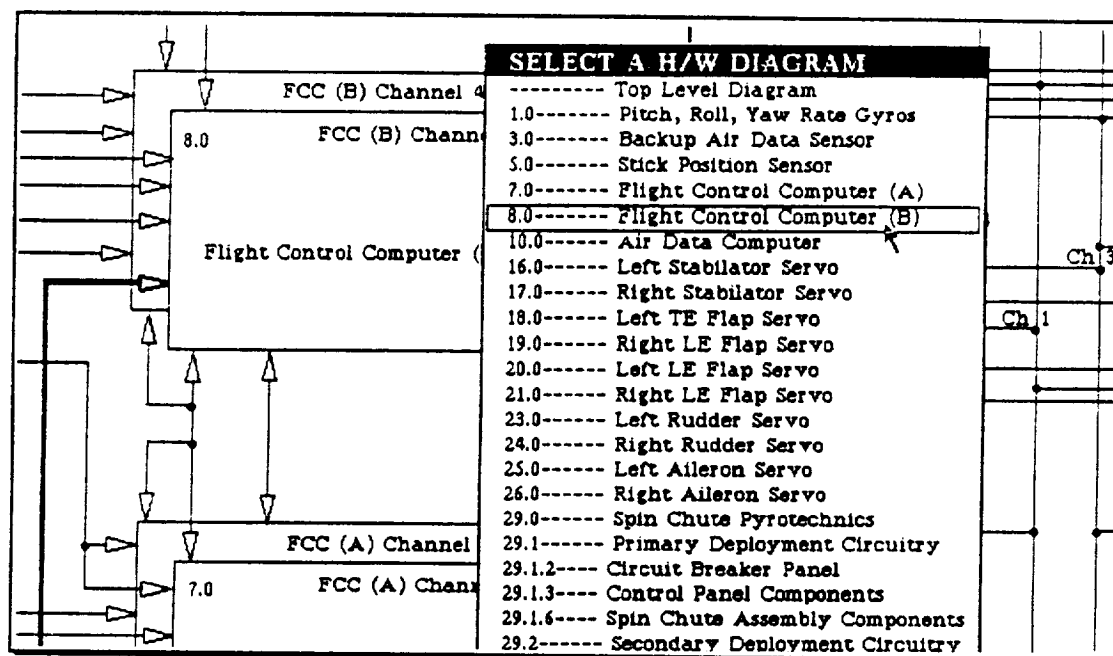


Figure 3.1.2-24. H/W Diagram Selection Menu

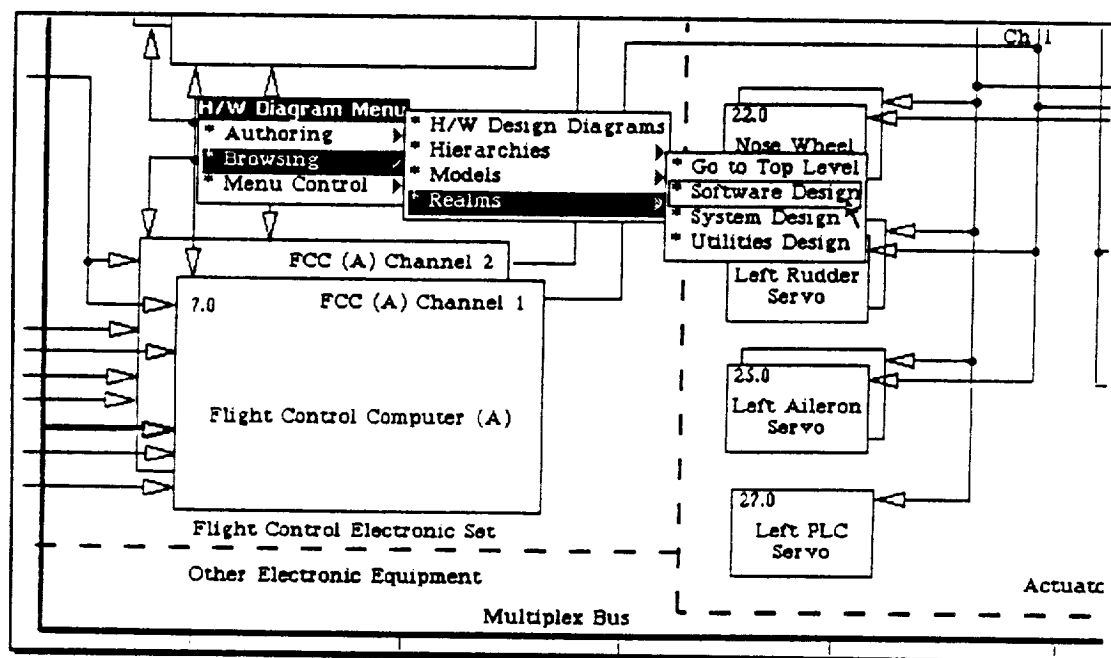


Figure 3.1.2-25. Realm Selection Browsing Options



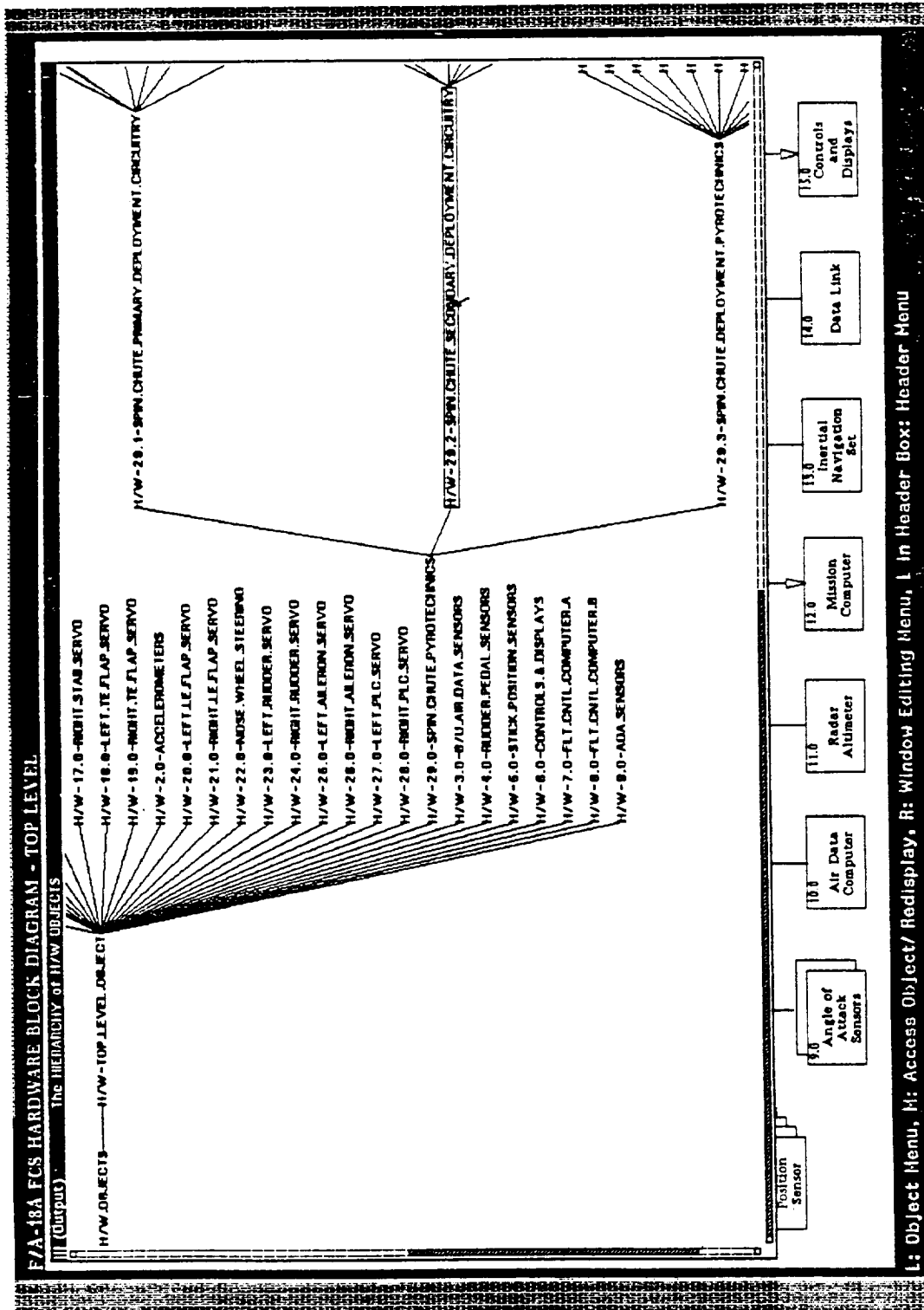


Figure 3.1.2-26. H/W Object Hierarchy (Decomposition) Display

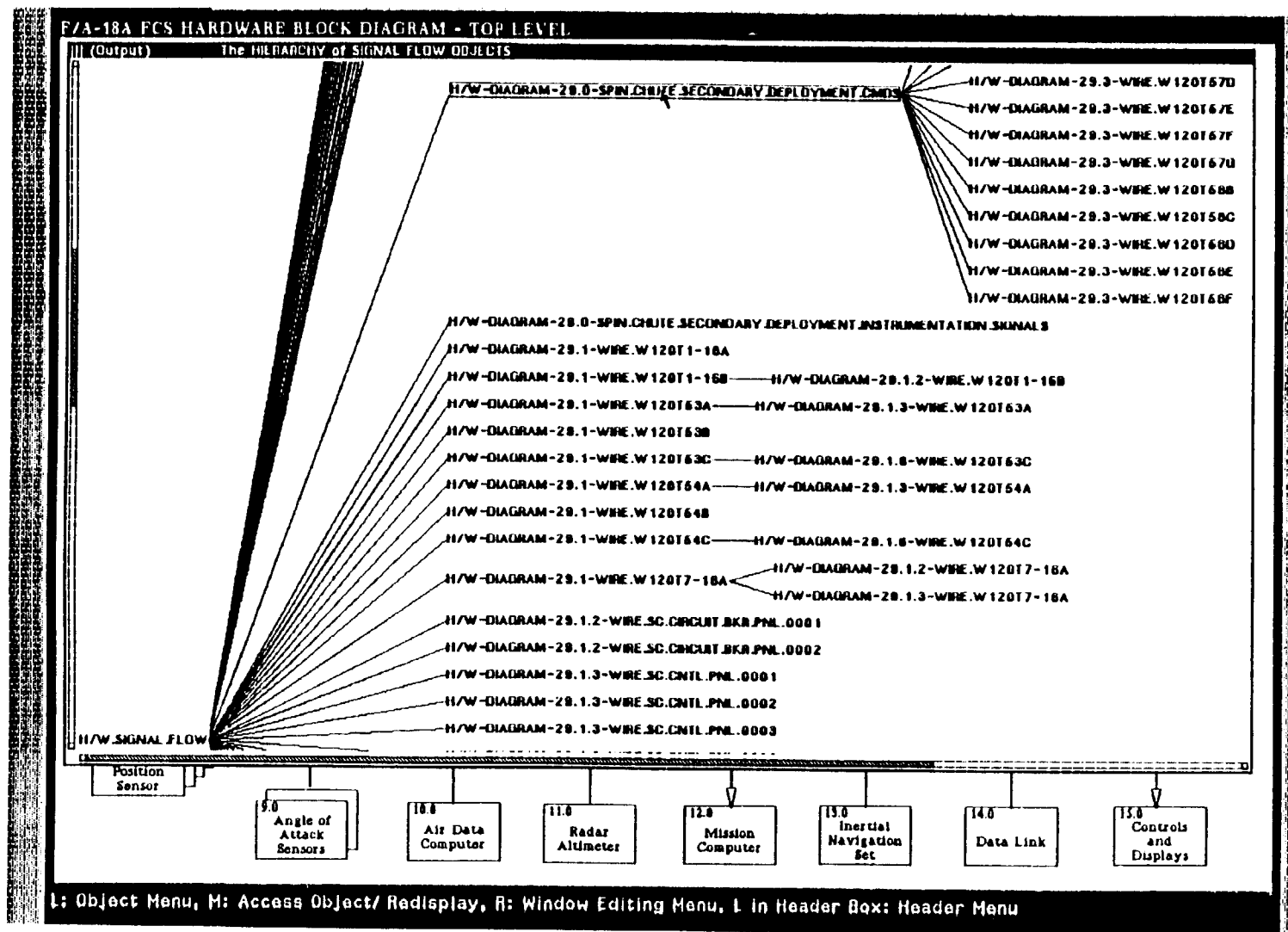


Figure 3.1.2-27. Signal Flow Hierarchy (Decomposition) Display

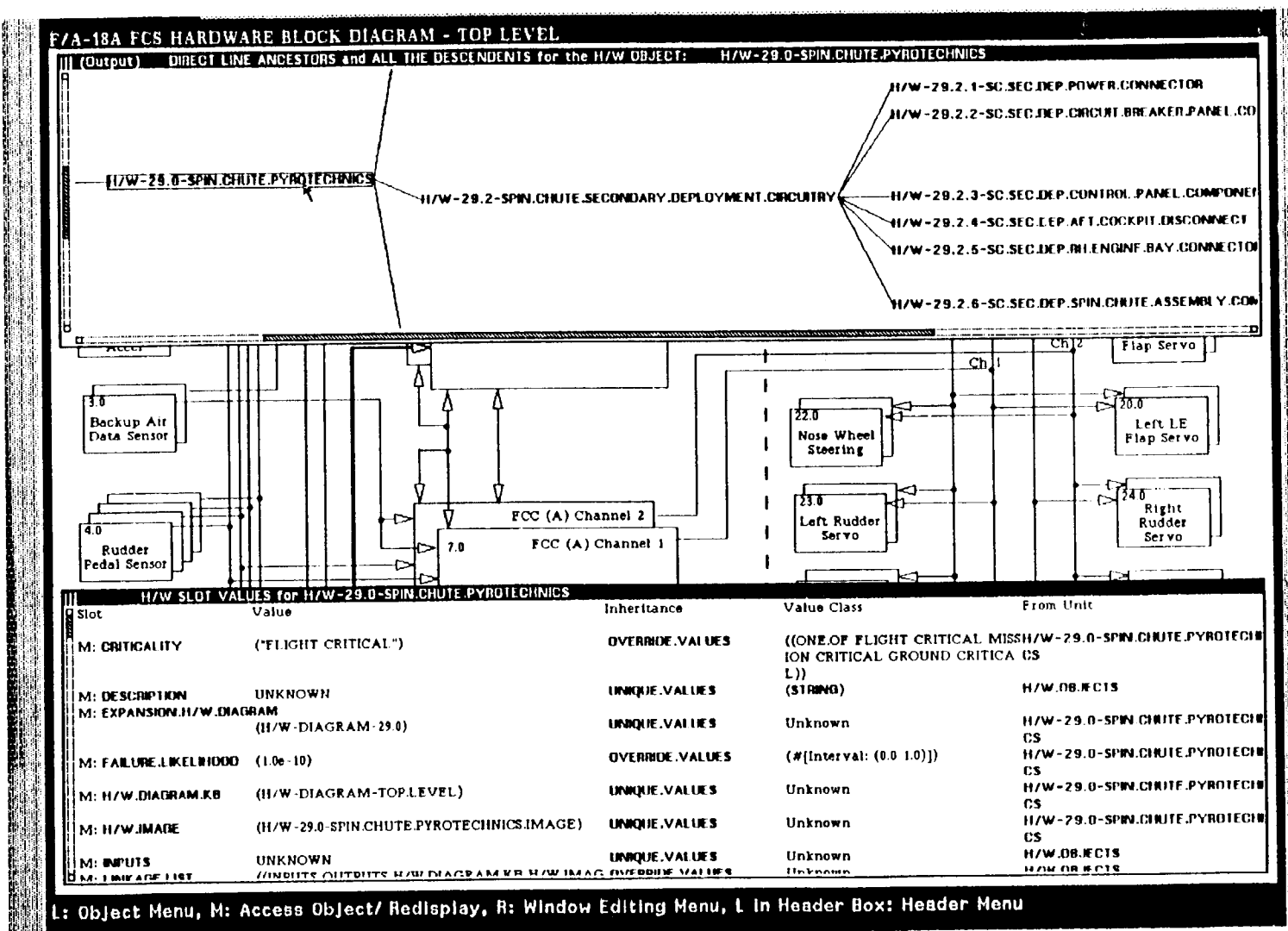
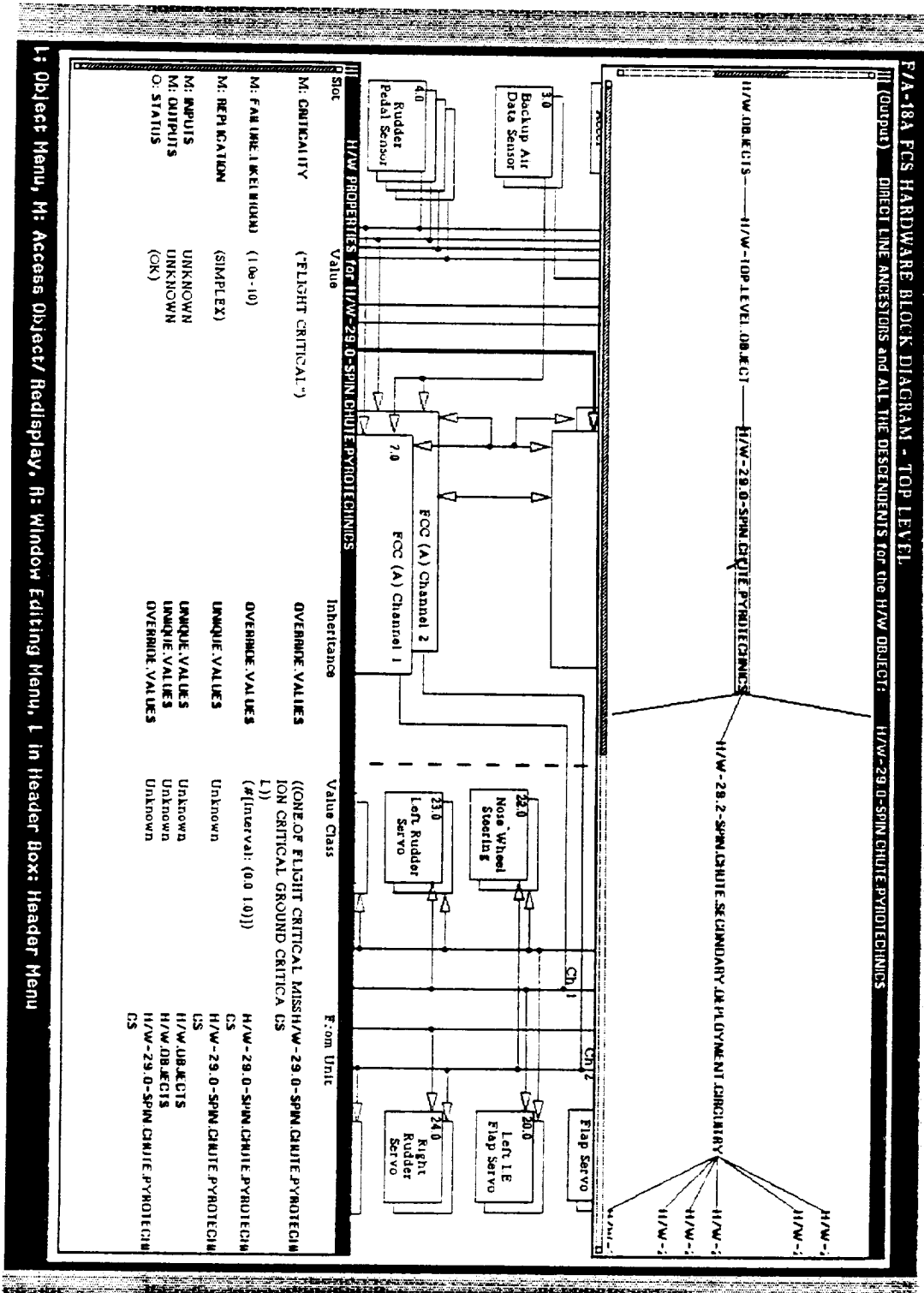


Figure 3.1.2-28. An All Properties Display for a H/W Object



**Figure 3.1.2-29. An FCS Properties Display for a H/W Object**

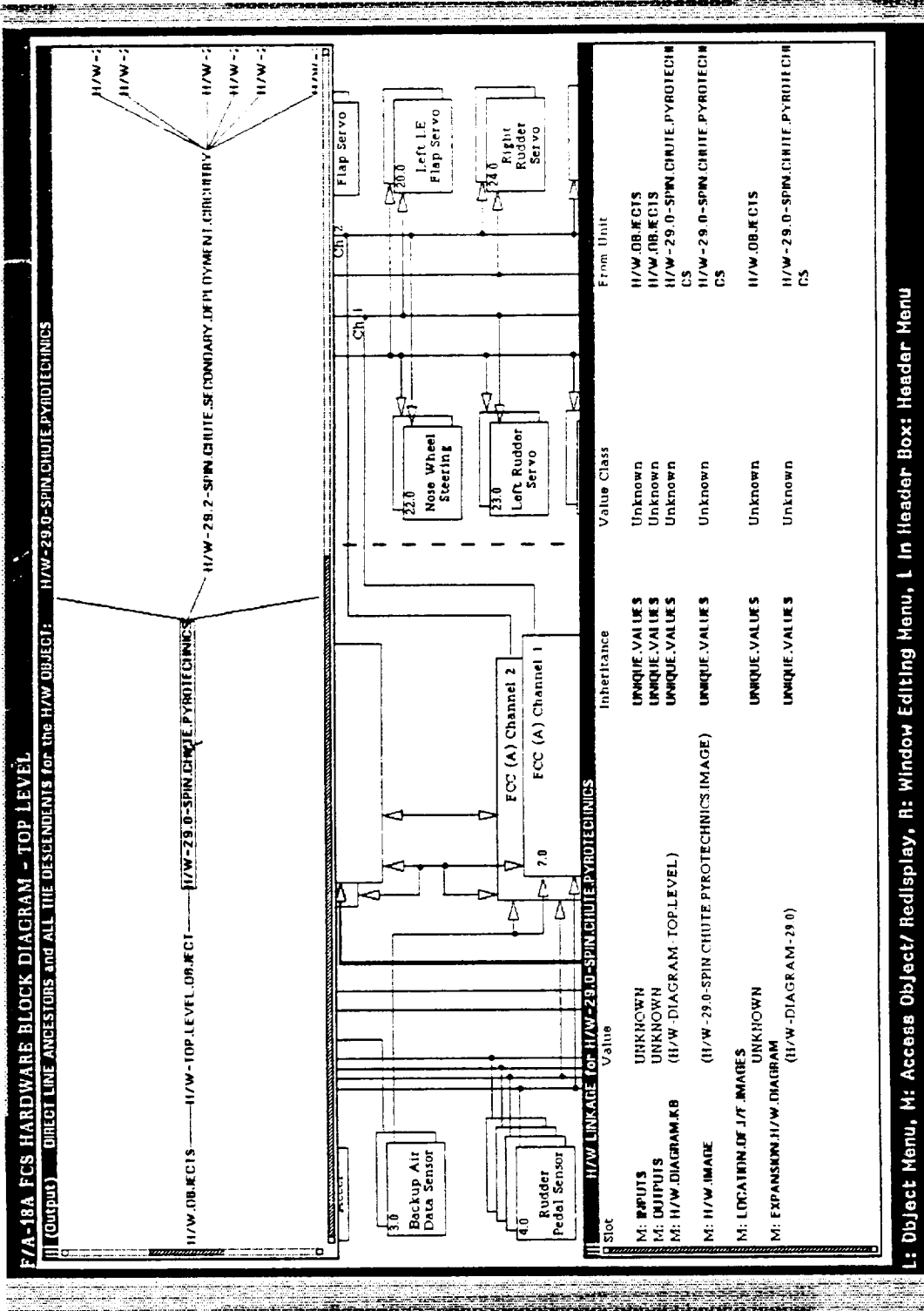
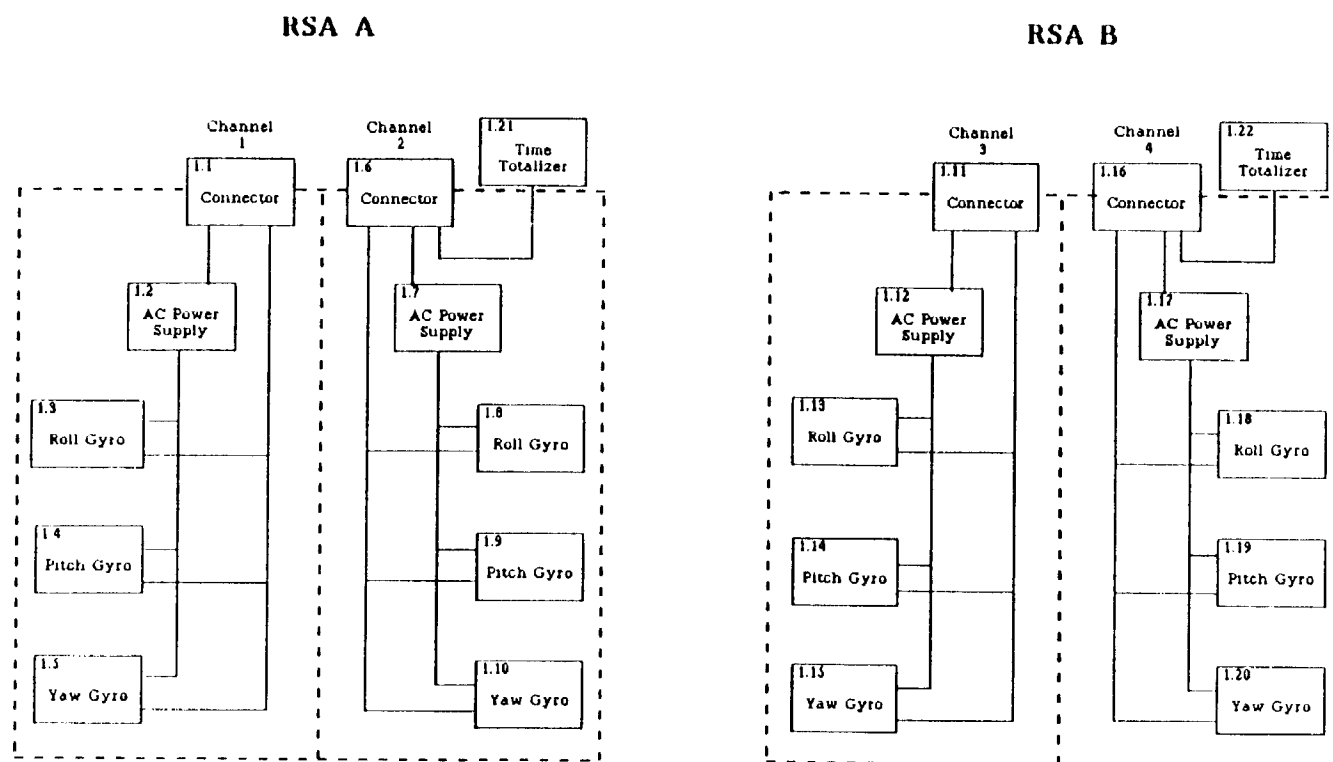


Figure 3.1.2-30. A KBS Linkage Properties Display for a H/W Object



RATE SENSOR ASSEMBLY FUNCTIONAL DIAGRAM

Figure 3.1.2-31. A Typical Expansion Display for a H/W Object

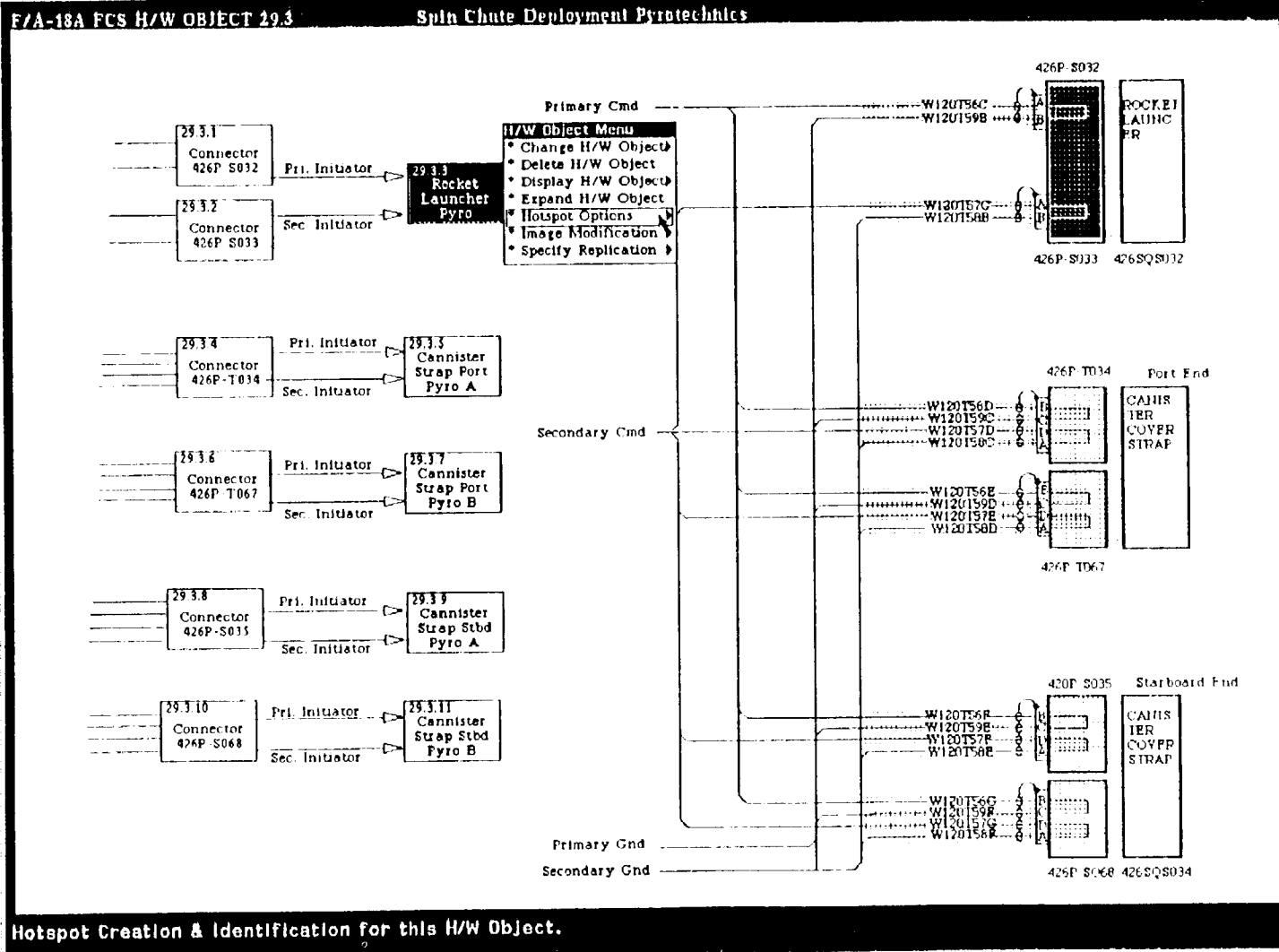


Figure 3.1.2-32. Hotspot Linkage Highlighting

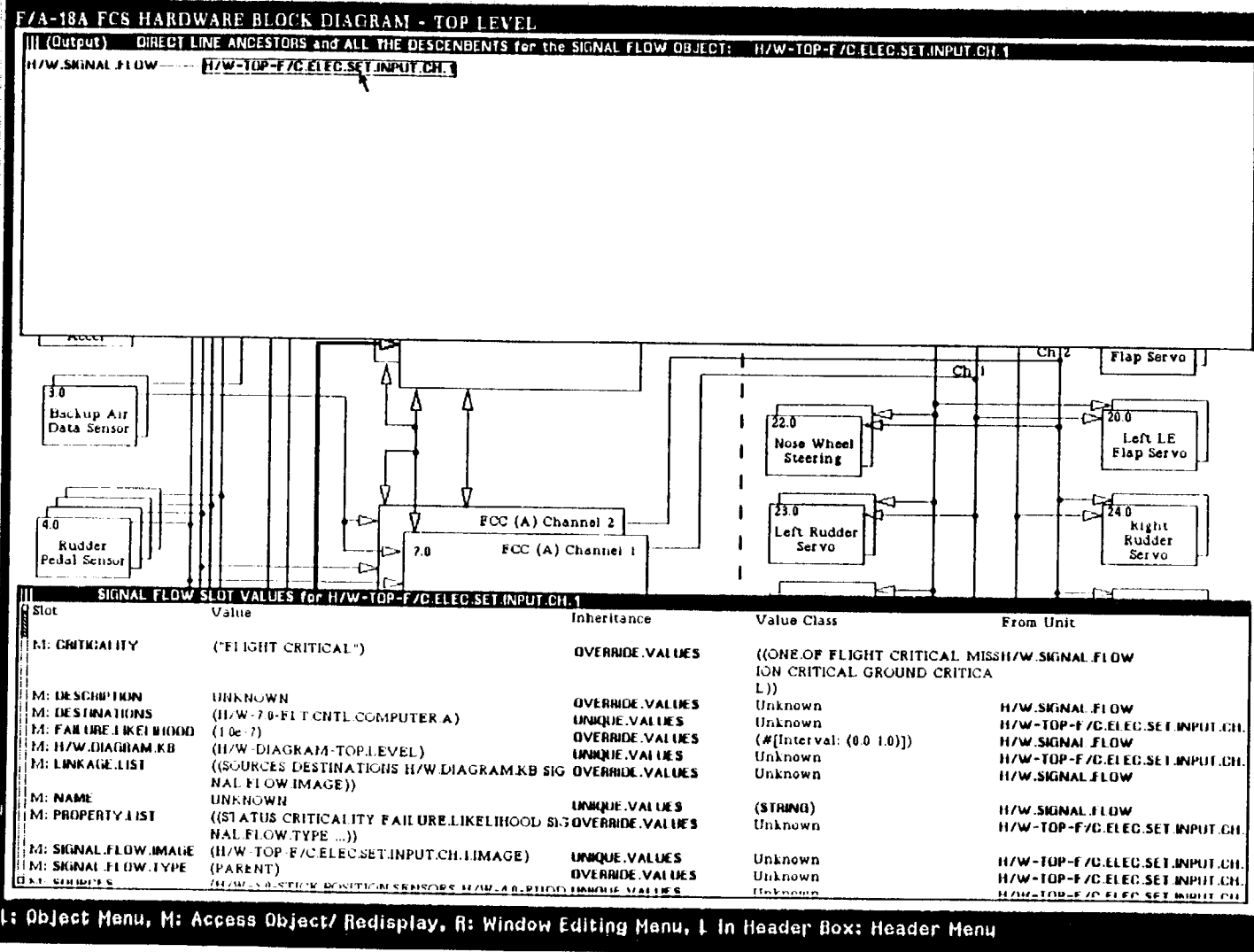


Figure 3.1.2-33. An All Properties Display for a S/F Object



### 3.1.3 Validation Concepts

The semantic network functionality is dependent upon the existence of a *valid* linkage within the network. The authoring methods help the user create objects with valid linkage and allow the user to modify objects in a valid fashion. In addition, validation methods have been provided which enable the user to test these linkages within the network for the hierarchies which utilize the SA methodology.

These methods enable the user to validate that the links between objects and their images are valid. This validation proves that all images know who their object is, that all objects know who their image is and that this linkage is consistent. Namely, that each object and its image point at one another.

In addition, it is possible to validate that the connectivity is valid. The object connectivity validation methods prove the object outputs correspond with data flow sources and that object inputs correspond with data flow destinations. In addition, these validation methods verify that every data flow object has one source and one destination.

Figure 3.1.3-1 depicts the validation object connectivity methods which are provided in the DFD menu. When the *Object Connectivity* validation option is selected, another menu of validation options is displayed (Figure 3.1.3-2). This second menu allows the user to select the type of objects (process, data flow, external or all) to be validated. Figure 3.1.3-3 depicts the results of the connectivity validation of all of the objects on a single DFD, the Top Level DFD in this case. Figure 3.1.2-4 depicts the results of the connectivity validation of all of the objects on all of the DFDs. In this case, 7 errors are found in which data flow stubs have not been connected. These errors indicate that the system design decomposition is incomplete.

The results of an *Image Links* validation option selection for a single DFD and for all of the system design KB are depicted in Figures 3.1.3-5 and 3.1.3-6 respectively.

## Knowledge Capture Concepts

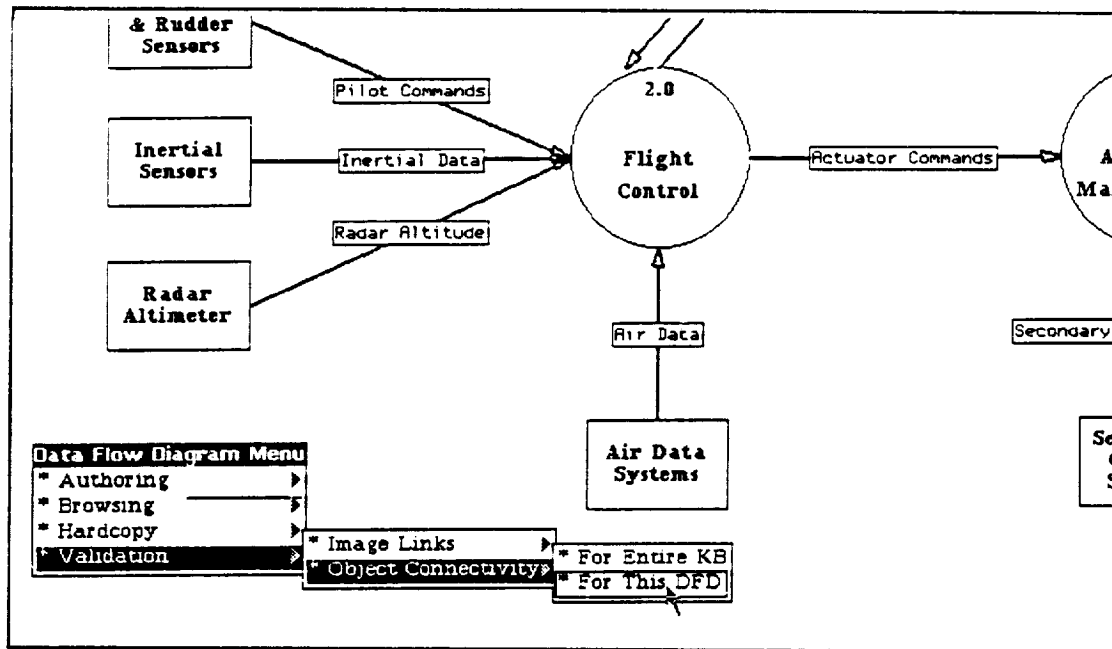


Figure 3.1.3-1. DFD Menu Validation Options

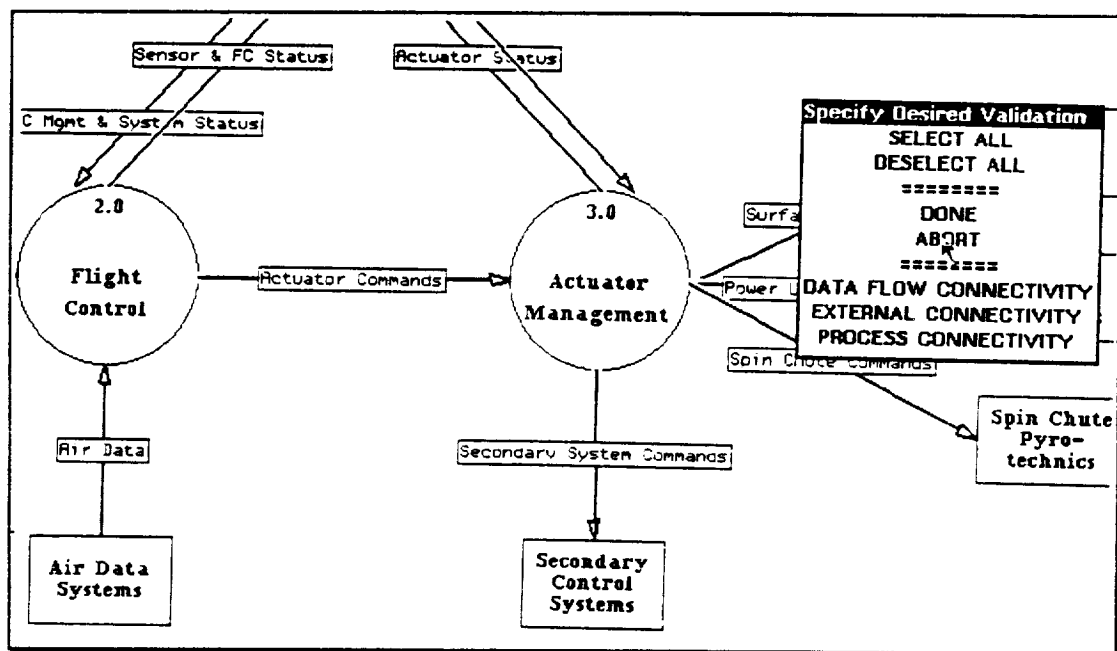
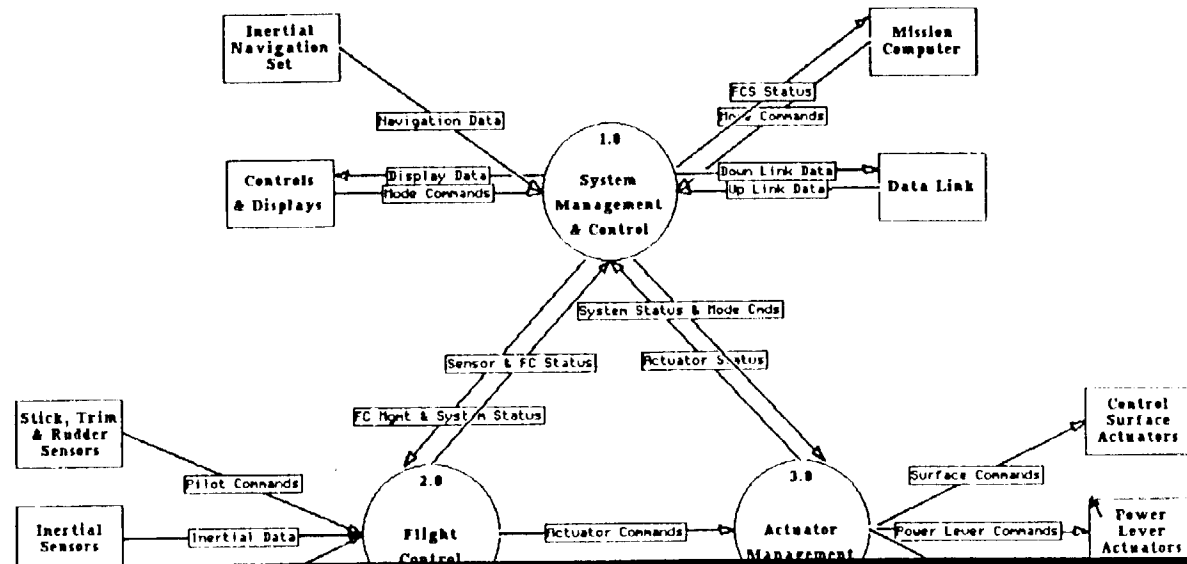


Figure 3.1.2-2. Menu of Connectivity Validation Options

F/A-18A FCS TOP LEVEL DFD



Connectivity Validation for F/A-18A SYSTEM-DFD-TOP.LEVEL System Design Objects

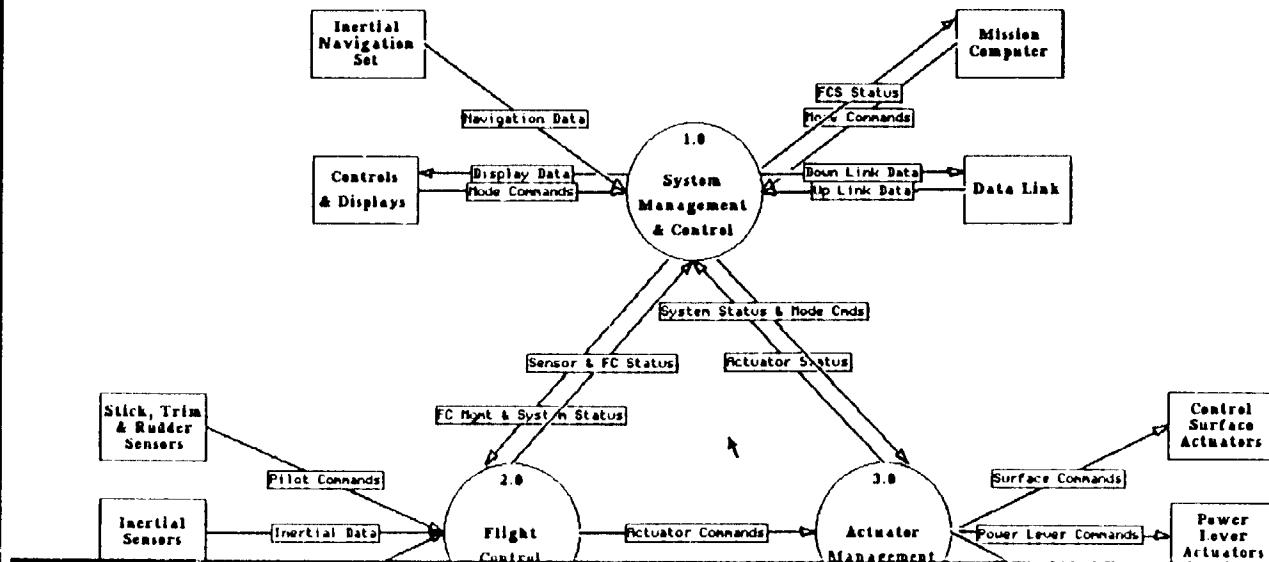
-----> STARTING a CONNECTIVITY CHECK for the 20 DATA FLOW OBJECTS specified in SYSTEM-DFD-TOP.LEVEL.  
The CONNECTIVITY CHECK for the 20 SYSTEM-DFD-TOP.LEVEL DATA FLOW OBJECTS is COMPLETE. ----> 0 ERRORS ENCOUNTERED

-----> STARTING a CONNECTIVITY CHECK for the 12 EXTERNAL OBJECTS specified in SYSTEM-DFD-TOP.LEVEL.  
The CONNECTIVITY CHECK for the 12 SYSTEM-DFD-TOP.LEVEL EXTERNAL OBJECTS is COMPLETE. ----> 0 ERRORS ENCOUNTERED

-----> STARTING a CONNECTIVITY CHECK for the 3 PROCESS OBJECTS specified in SYSTEM-DFD-TOP.LEVEL.  
The CONNECTIVITY CHECK for the 3 SYSTEM-DFD-TOP.LEVEL PROCESS OBJECTS is COMPLETE. -----> 0 ERRORS ENCOUNTERED

Figure 3.1.3-3. Connectivity Validation Display for All of the Objects Created on One DFD

## F/A-18A FCS TOP LEVEL DFD



## Connectivity Validation for F/A-18A System Design Objects

```

-----> STARTING a CONNECTIVITY CHECK for all 98 of the DATA FLOW OBJECTS.
ERROR: The data flow object DFD-2.2.2-AFC.MODE.CMDS has no destination.
ERROR: The data flow object DFD-2.2.1-PFC.MODE.CMDS has no destination.
ERROR: The data flow object DFD-2.2.3-SSC.MODE.CMDS has no destination.
ERROR: The data flow object DFD-2.2.2-AFC.STATUS has no source.
ERROR: The data flow object DFD-2.2.1-PFC.STATUS has no source.
ERROR: The data flow object DFD-2.2.3-SSC.STATUS has no source.
ERROR: The data flow object DFD-3.4-SPIN.CHUTE.ACTUATOR.STATUS has no source.
The CONNECTIVITY CHECK for all 98 of the DATA FLOW OBJECTS is COMPLETE. ----> 7 ERRORS ENCOUNTERED

-----> STARTING a CONNECTIVITY CHECK for all 12 of the EXTERNAL OBJECTS.
The CONNECTIVITY CHECK for all 12 of the EXTERNAL OBJECTS is COMPLETE. ----> 0 ERRORS ENCOUNTERED

-----> STARTING a CONNECTIVITY CHECK for all 28 of the PROCESS OBJECTS.
The CONNECTIVITY CHECK for all 28 of the PROCESS OBJECTS is COMPLETE. ----> 0 ERRORS ENCOUNTERED
  
```

Figure 3.1.3-4. Connectivity Validation Display for All of the Objects in the System Design Realm

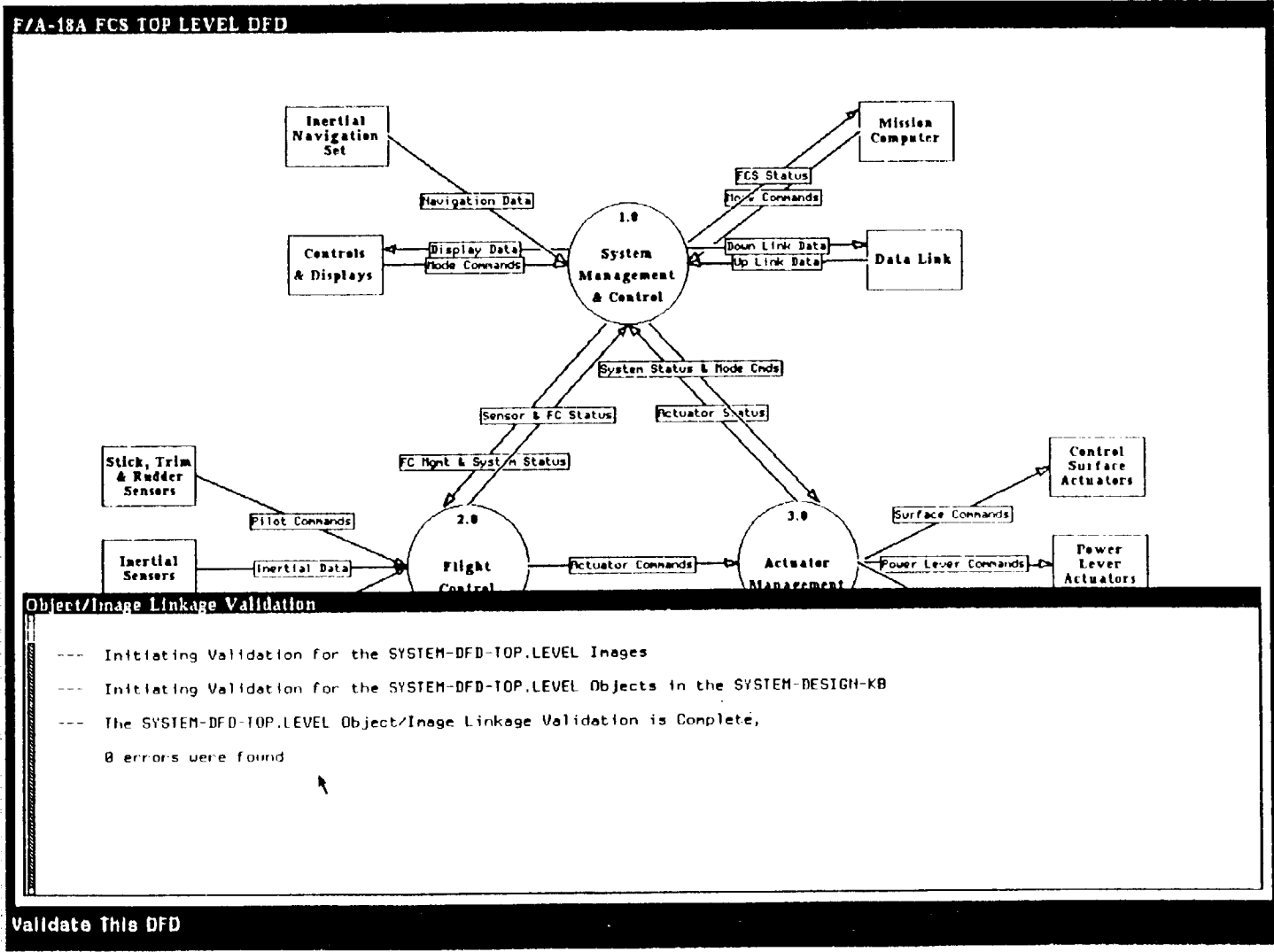
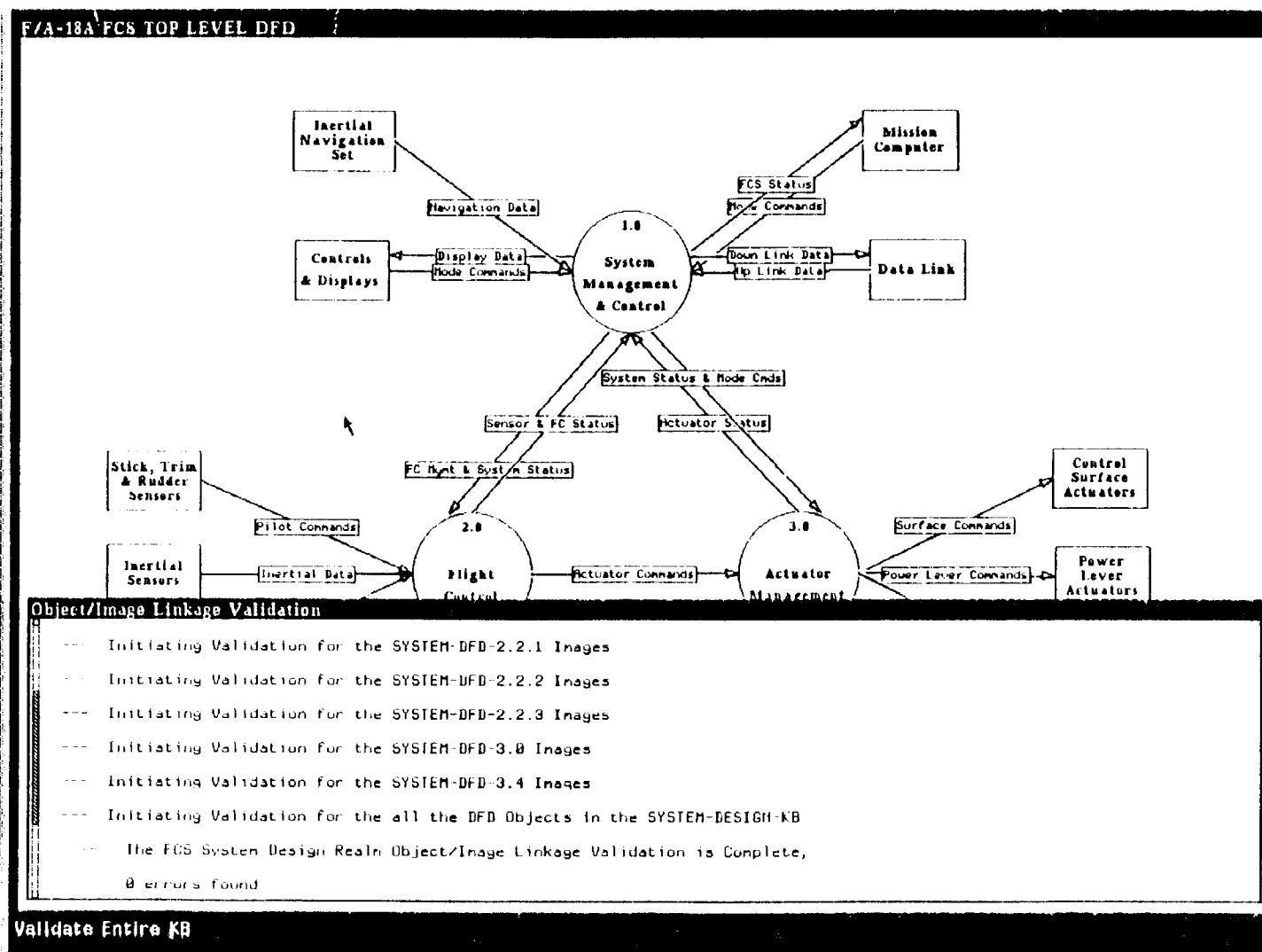


Figure 3.1.3-5. Image Link Validation Display for All of the Images on One DFD



**Figure 3.1.3-6. Image Link Validation Display for All of the DFD Images in the System Design Realm**

### 3.2 Bottom Up Schematic Capture

The design process is frequently a combination of top down decomposition and bottom up aggregation. These two design processes are viewed as complementary. In the bottom up aggregation approach, the subsystem design is incorporated into the larger picture of a total system. Today's technology provides CAD systems which are dedicated to the design of subsystems, both mechanical and electrical. The schematic capture mechanisms are intended to incorporate CAD subsystem designs, as reflected in schematics, into the semantic network which captures the entire system design.

These schematic capture mechanisms have intentionally been restricted to utilize source knowledge obtained by scanners. This restriction largely reflects resource constraints. It also allowed the effort to focus on the schematic capture issues relevant to bottom up knowledge capture and to side step the resource consuming process of integrating multiple platforms, CAD systems and operating environments. Thus the schematic capture process described here starts with hardcopy schematics which are scanned and captured as bitmapped images. These bitmaps are used by the KCS as the graphical input to the schematic capture process.

Mechanisms are then provided for the user which enable individual objects to be identified (from the schematic) as H/W objects or as S/F objects in the schematic network. Objects which have been obtained from the schematic are treated as objects on the lowest level H/W diagram in a decomposition hierarchy. Other mechanisms are provided which enable the user to aggregate the low level objects into the next level up of a H/W diagram hierarchy. The final intent is to provide mechanisms which merge the top down decomposition objects and bottom up aggregation objects into a common hierarchy.

The Utility Design Realm H/W diagram menu has been augmented with a *Bottom Up Design* item which provides access to the bottom up design capture exploratory mechanisms and the selected HARV design data which has been captured with this facility. Figure 3.2-1 depicts the *Bottom Up Design* menu item and its cascading options. This menu item provides access to the Emergency Hydraulic Model described in Sections 4.4 and 5.3, the Emergency System Schematics and the Sample Motor Control Model described in Sections 3.2 and 3.3.

## Knowledge Capture Concepts

The Emergency System Schematics are all accessible from the Utility Design Realm H/W diagram menu. Figure 3.2-2 depicts one quadrant of one of the 11"x17" hardcopy schematics scanned into the KCS. Scrolling and zoom mechanisms are available to browse these schematics. Four of the nine schematics have been copied to form one collection of schematics to form the schematic layer for the Emergency Hydraulic Model bottom up aggregation which is discussed and illustrated in Sections 4.4 and 5.3.

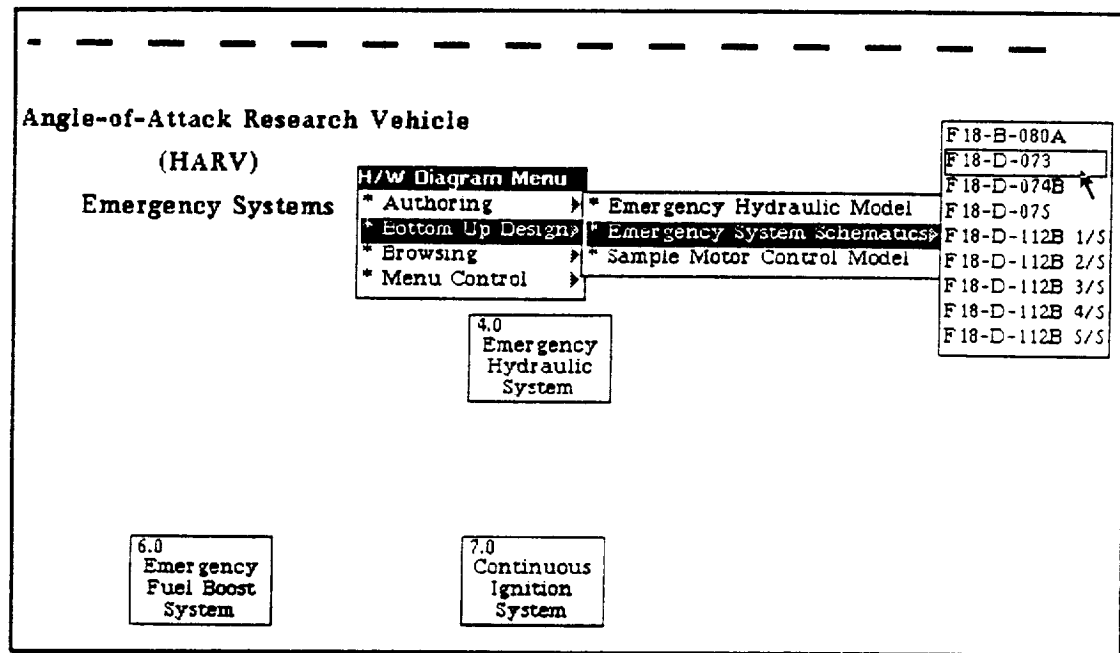


Figure 3.2-1. The Utility Design Realm *Bottom Up Design* Cascading Menu Items

A simple motor control circuit has been used to develop concepts and to demonstrate the nature of schematic capture. This schematic capture capability was developed by Scott Sikora as part of his thesis work<sup>7</sup>. Figures 3.2-3 through 3.2-6 depict how a bottom level design can be captured, aggregated and built upward with the goal of merging with a top down design decomposition. This simple motor control circuit was scanned into the KCS and is illustrated in Figure 3.2-3.

Figure 3.2-3 also illustrates the *Bottom Up: Schematic Menu*. The *Schematic Capture* item on the menu allows the user to capture objects from bitmap graphics. The mechanism allows the user to identify objects on the bitmap with the mouse. Rectangular areas are identified by the user and are highlighted by the KCS. The *Menu Control* item gives the



## Knowledge Capture Concepts

user access to the KEE™ functionality. The remaining menu items primarily consist of browsing mechanisms, e.g. the *Change Level* item allows the user to browse the Utilities Design Realm bottom up design diagrams. Similarly, the *Utilities Realm* item allows the user to select the top level Utilities Design Realm H/W diagram and returns the user to the top down decomposition diagrams.

Figure 3.2-4 indicates the bottom level schematic diagram that is generated after the objects, object types and connections are defined by the user. Note the use of highlighting used to indicate the objects identified by a user. This figure also indicates the *Object Operations* menu used to manipulate the objects. The *Associate Another Image* item allows the user to associate multiple image rectangle selections with a single object. The other items allow the user to browse and modify the object with the *Change Status*, *Delete*, *Display Object* and *Rename* command items.

Figure 3.2-5 depicts the bottom level object abstraction developed from the user's selection of objects, depicted in Figure 3.2-4. Figure 3.2-6 depicts how an aggregation can be performed on the bottom level abstraction. The access to the functionality which enables the user to aggregate and group objects is provided in the *Bottom Up: Abstract Menu* depicted in Figure 3.2-5. Here, the *Insert a New Level 0* menu item found on the bottom level diagram menu was used create a new upper level diagram. Upper level diagrams are designated as level 0 in the bottom up aggregation process. The *Group Objects* menu item was then used to group the POWER, WIRE.A and SWITCH objects to form the CONTROL object depicted in Figure 3.2-6.

The menu items depicted in Figures 3.2-5 and 3.2-6 provide rule creation and modeling mechanisms in addition to the authoring and browsing mechanisms used for bottom up aggregation. The rule capture and modeling mechanisms are described in Section 3.3.



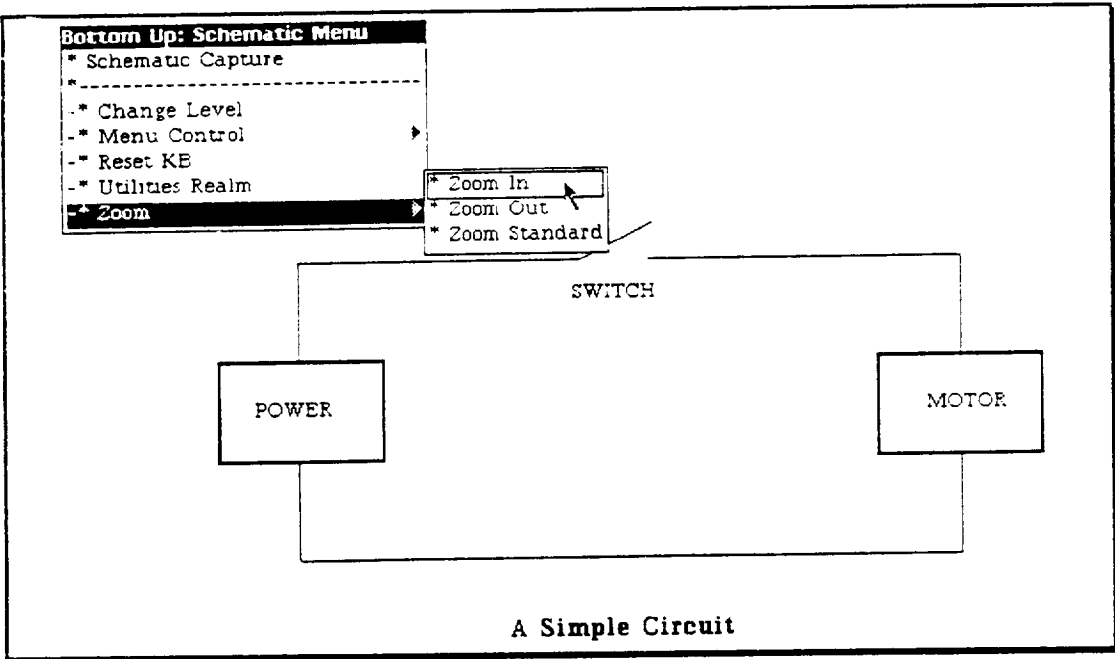


Figure 3.2-3. Motor Control Schematic with the *Bottom Up Schematic Menu*  
- A Demonstration Bitmap Graphic -

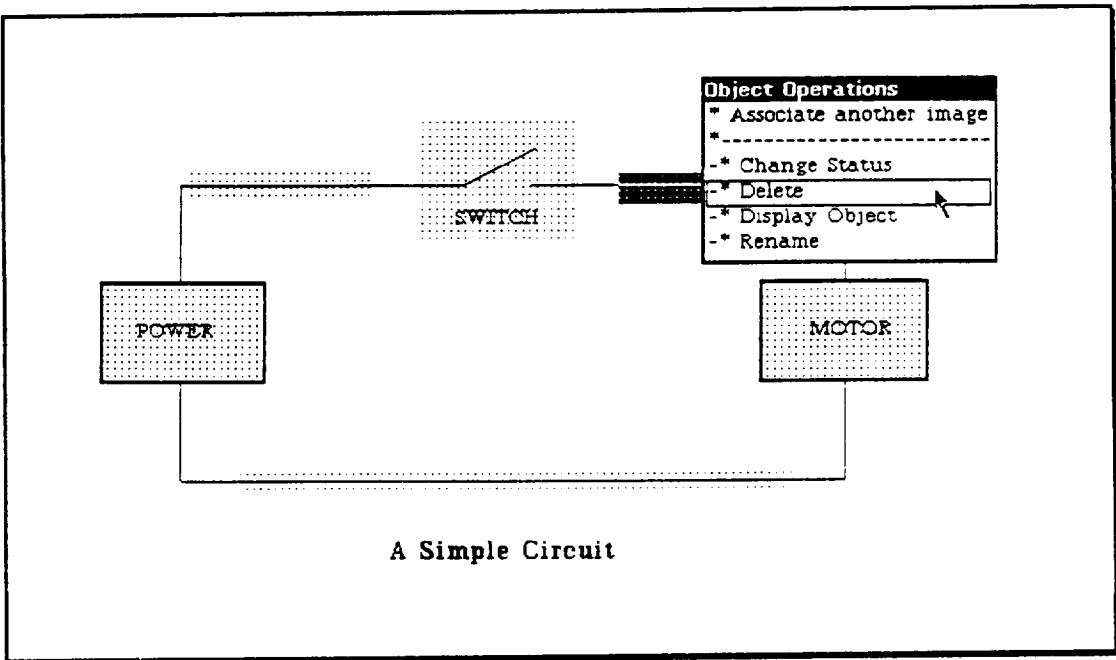


Figure 3.2-4. Motor Control Schematic with the *Object Operations Menu*  
- A Demonstration Bitmap Graphic with Object Identification Highlighting -

## Knowledge Capture Concepts

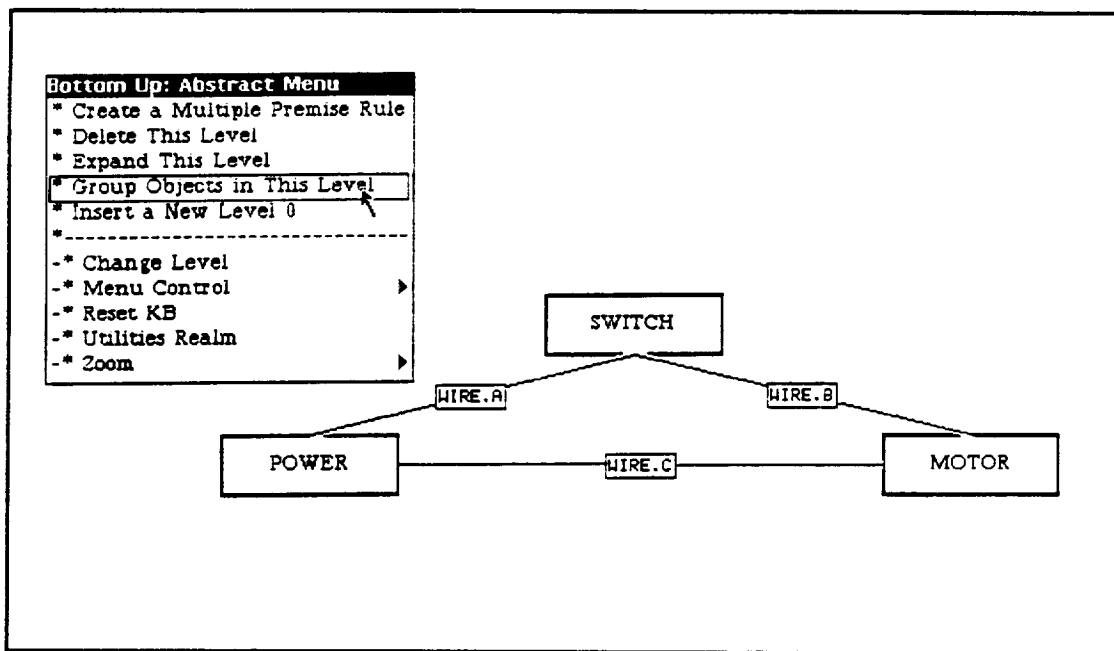


Figure 3.2-5 Bottom Level Abstraction of the Demonstration Motor Control Schematic with the *Bottom Up: Abstract Menu* - A Level 1 Abstraction in this Case

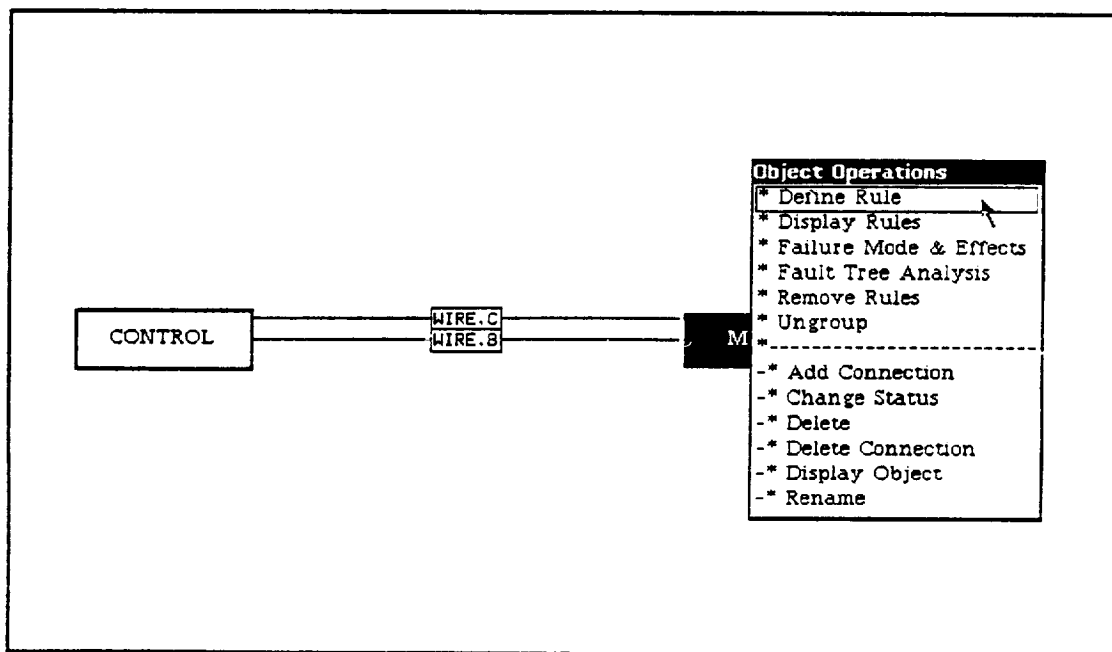


Figure 3.2-6. An Aggregation Example Using the Demonstration Motor Controller with the *Object Operations* Menu - A Level 0 Abstraction in this Case

### **3.3 Rule-Based Models**

The top down decomposition and the bottom up schematic capture mechanisms are designed to create objects which are compatible with an inference engine. The coupling of the semantic network design knowledge encryption with an inference engine in this problem domain supports the development of *dynamic models* which can be used to bring the design knowledge to life. The wide variety of useful models possible in this KCS environment are envisioned as a source of insight that is necessary and currently unavailable in this problem domain.

#### **3.3.1 Exploratory Behavioral Models**

It should be recognized that the ensuing discussion of exploratory rule-based models for the FCS (and, more broadly, the general case of automated systems) problem domain is limited. The resources allocated to this project have permitted the exploration of only a small sample of the considerable modeling potential offered by the KCS environment. These resources have been focused on a set of illustrative models. The general goal of the selected models was that they should be of a behavioral nature. They should answer the thrust of the question "*How does this design/device work?*"

##### 3.3.1.1 Operational Models

The way in which the FCS functions is dependent upon a number of criteria which include the aircraft state, the status of the FCS hardware, the current operational mode and pilot commands. This functionality is sufficiently complex that behavioral models are viewed as being useful to a number of potential users throughout the aircraft's lifetime. The knowledge embedded in the semantic network contains the interdisciplinary knowledge necessary to implement such behavioral models.

When the state, mode, hardware status and command knowledge is combined with *if..then..* rule-based models to define the functional operation of the FCS, it is possible to implement data driven operational models with the inference engine. Changes in state, mode, hardware status and commands will cause the rule-based models to draw different conclusions regarding the FCS which can be displayed to the user. These displays can be designed to indicate the causal nature of the associated FCS criteria. Such models are

## Knowledge Capture Concepts

dynamic and can be used to explore the interaction of multiple criteria upon the functioning of the FCS.

One model has been implemented which indicates the response of the FCS to changes in state, mode and pilot commands. This model is centered on the operation of the nose wheel steering and is focused on a display of the control block diagram for this aircraft system. It illustrates the active control path in the diagram based upon discrete criteria which can be modified by the user. This dynamic interactive model is described in detail in Section 5.1.

### 3.3.1.2 Failure Mode Models

One significant FCS design consideration that lends itself to rule-based modeling is associated with the effects of failures. In this problem domain, Failure Modes and Effects Analyses (FMEAs) typically study the effects of a component's failure and Fault Tree Analyses (FTAs) typically identify the ways in which a given device (or capability) can fail.

The behavior of a device (or a capability) can be defined as a set of *if...then...* rules which include hardware component status criteria in their premises. This hardware component status is available from the H/W and S/F objects defined with KCS authoring mechanisms in the H/W design realm. When these rules are activated with the inference engine forward chaining mechanism (data driven), it is possible to display the results in a form generally associated with an FMEA study. Furthermore, when the rule set utilizes contexts (or other rule execution control mechanisms) which focus on specific aspects of the device functionality, the single rule set can be used to create more than one FMEA. Thus, one rule set will support multiple FMEAs.

Since the behavioral rules utilize components defined in the semantic network, it is possible to utilize the KCS browsing mechanisms to peruse the device and to selectively fail its components (one or more) interactively. This interactive selection of failed components enables each FMEA to be executed with a diverse set of component failures.

When these rules are activated with the inference engine backward chaining mechanism (goal driven), it is possible to display the results associated with an FTA study. Note that the FTA rule set can be the same one developed for the FMEAs. Here again this single rule

## Knowledge Capture Concepts

set, if it uses mechanisms which identify specific aspects of a device's design, can be used to generate more than one FTA.

When failure analyses are performed manually, they are sufficiently time consuming that their number and scope are restricted by cost considerations. The rule-based automation of these failure analyses should reduce the cost of these failure analyses. A more significant advantage is seen in the relative ease with which a rule set may be modified. Such modification would enable multiple design alternatives and subsequent design changes to be examined. In contrast, multiple manual failure analyses for the purposes of design modification analysis or design change analysis can be prohibitively expensive.

Two aircraft systems have used to explore the use of failure models. One system, the spin chute deployment system (necessary for test pilot survival), uses a rule-based failure model based upon objects decomposed in the H/W realm to the component level. The spin chute deployment model is discussed in detail in Section 5.2. The other system, the emergency hydraulic system (also necessary for test pilot survival) uses a rule-based model based upon objects captured from a schematic. The emergency hydraulic system model is discussed in detail in Section 5.3.

### **3.3.2 Behavior Definition**

The schematic capture mechanism has been enhanced with mechanisms which help the user perform FMEAs and FTAs. These mechanisms aid the user in specifying the behavior of the objects identified with the schematic capture mechanisms. They provide a *rule capture* capability which assists the user in specifying the appropriate behavioral rules necessary to support automated FMEAs and FTAs. The resultant rules utilize the inference engine to automatically generate the FMEAs and FTAs. In addition, a *library* is provided to assist the user in the specification of the object types.

These behavior definition mechanisms have been designed with the central goal of improving the productivity of the design engineer in the development of FMEAs and FTAs. These mechanisms are intended to speed the definition of the analyses, provide more confidence in their accuracy and support the ability to rerun them when design changes have been introduced. All of these enhancements are based on the use of automated mechanisms.

### Knowledge Capture Concepts

The concept of integrating schematic capture, rule capture and library objects was developed by Scott Sikora as a *Modeling Authoring System*<sup>7</sup> for safety analyses. All of the material in this section is based upon his thesis work on this topic.

#### 3.3.2.1 Rule Capture

A rule capture mechanism is provided which aids the user in defining the behavior of an object. This mechanism utilizes the known failure modes to assist the user in the generation of the behavior rules. An inference engine is then used by these rules to generate FMEAs and FTAs. Figure 3.3-1 depicts the *Object Operations* menu provided when the SWITCH H/W object had been moused. Figure 3.3-2 depicts the assistance provided when the user selects the *Define Rules* option. Here the user is prompted to select an object affected by a SWITCH FAILURE. In this case, the output wire was selected and the user is prompted in Figure 3.3-3 to select the new state for the wire from the known options provided in this model. Subsequently, the user is requested to define the behavior (rule) if the SWITCH has NO-SIGNAL. Figure 3.3-4 displays the rules selected with this rule capture mechanism.

Since rules have been specified for all the objects in this model, automated analyses can be requested for *every* object. Here, one FMEA and one FTA are depicted in Figures 3.3-5 and 3.3-6. These analyses are generated in real time when requested by the user. Thus the user is free to modify his design and interactively examine the effects with FMEA and FTA analyses.

The resource constraints for this project made it necessary to focus this effort with the intent of developing a working system. This focus enabled the development of these behavior modeling mechanisms and the demonstration of this simple model and of the somewhat more complicated model described in Section 5.3. It was found that this rule capture mechanism did indeed considerably enhance the user's ability to specify the behavioral rules and assess all the *modeled* failure mode effects on an object by object basis. It should be recognized that the success of this modeling effort was dependent upon restricting the nature and the number of the failure modes. Although it is recognized that additional work is required to scale this modeling effort up to industrial application quality, it is seen as unique and as possessing significant potential.



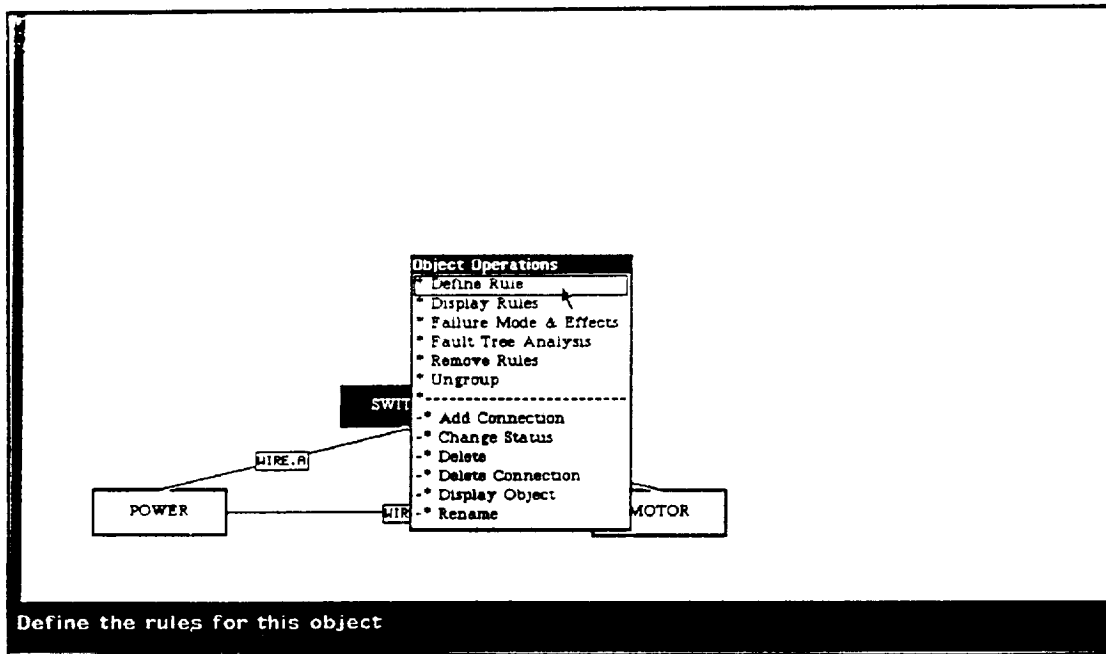


Figure 3.3-1. Rule Definition Request for a H/W object

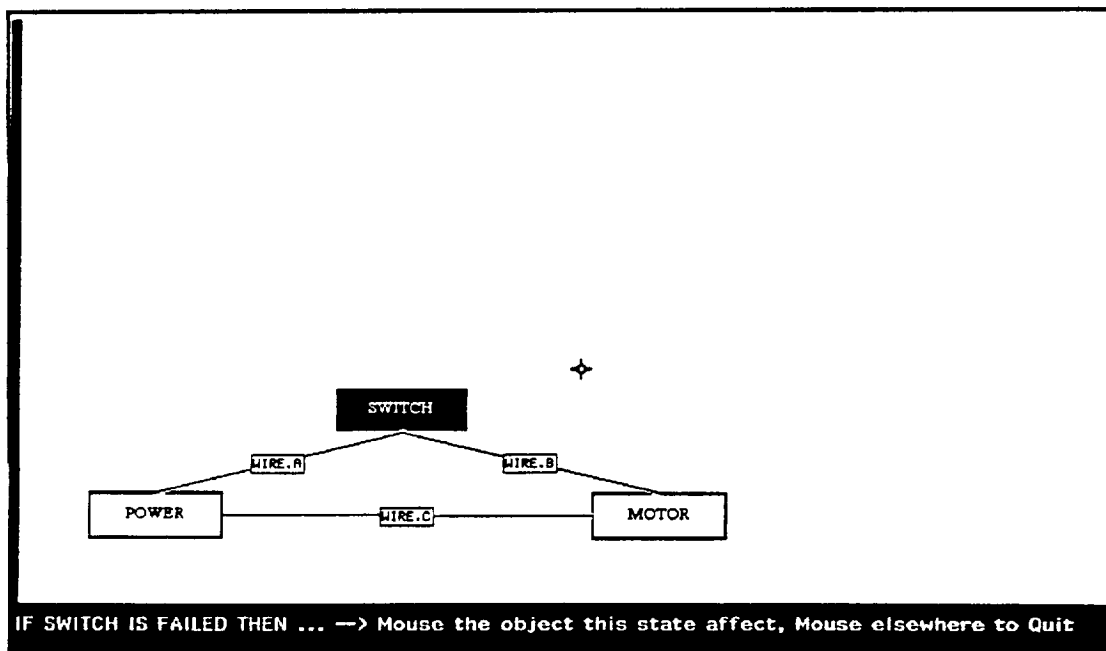


Figure 3.3-2. Rule Completion Assistance - Affected Object Selection

## Knowledge Capture Concepts

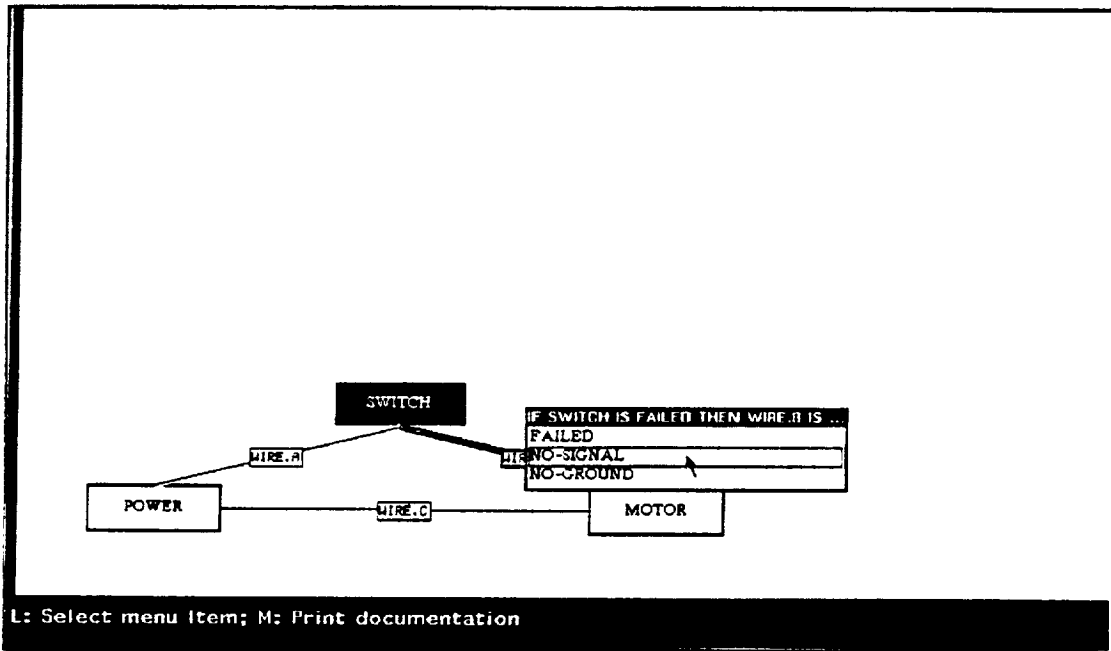


Figure 3.3-3. Rule Completion Assistance  
- Affected Object New State Selection -

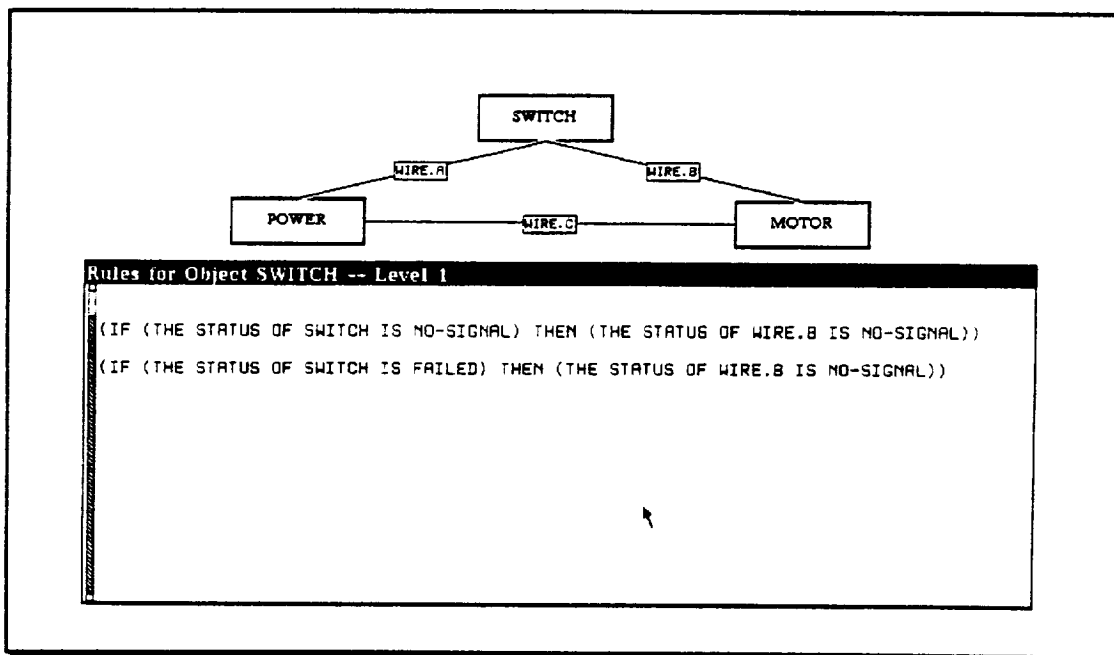


Figure 3.3-4. Rule Display for the SWITCH

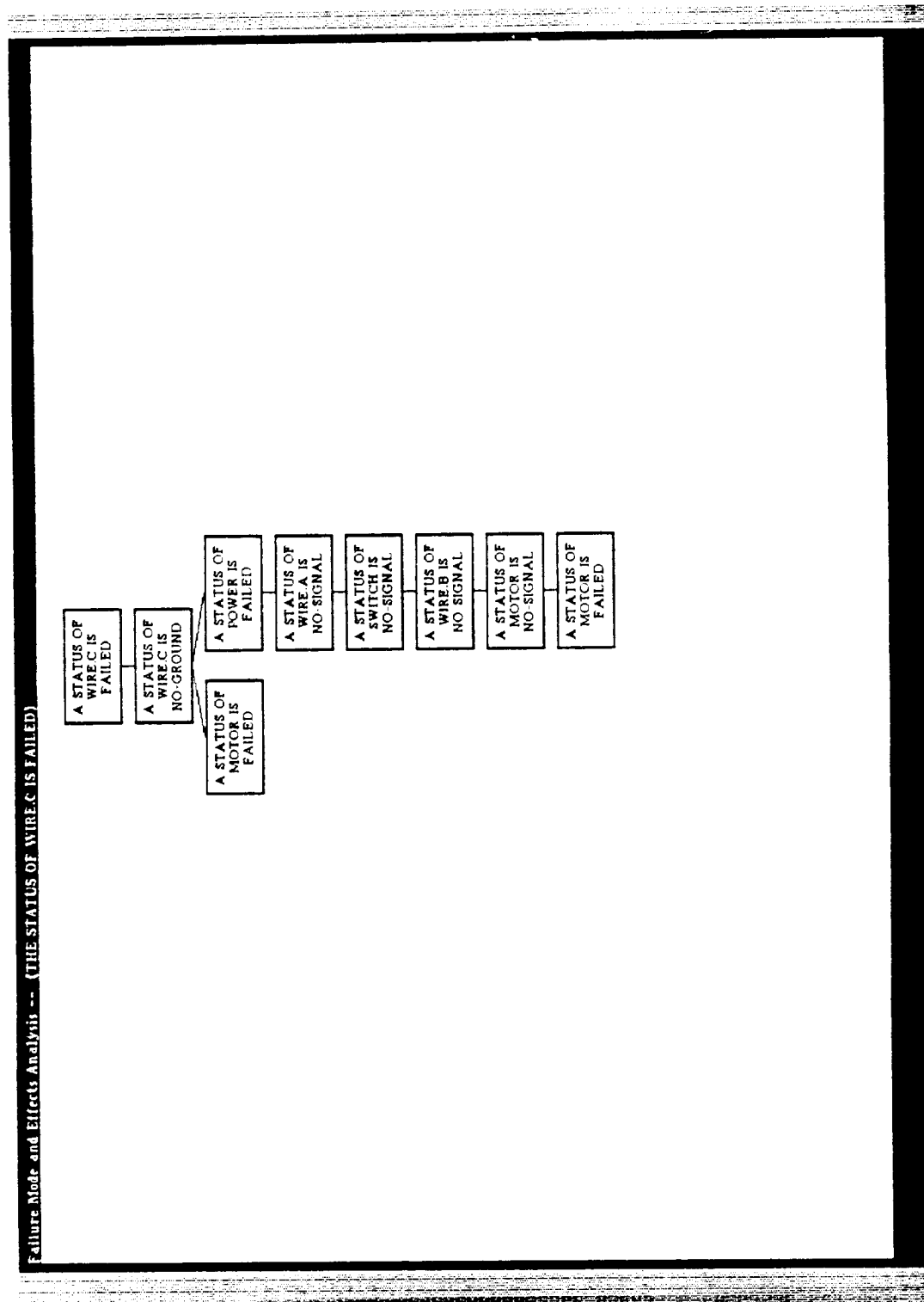


Figure 3.3-5. An FMEA Analysis for the Demonstration Motor Control Model - WIRE.C is Failed

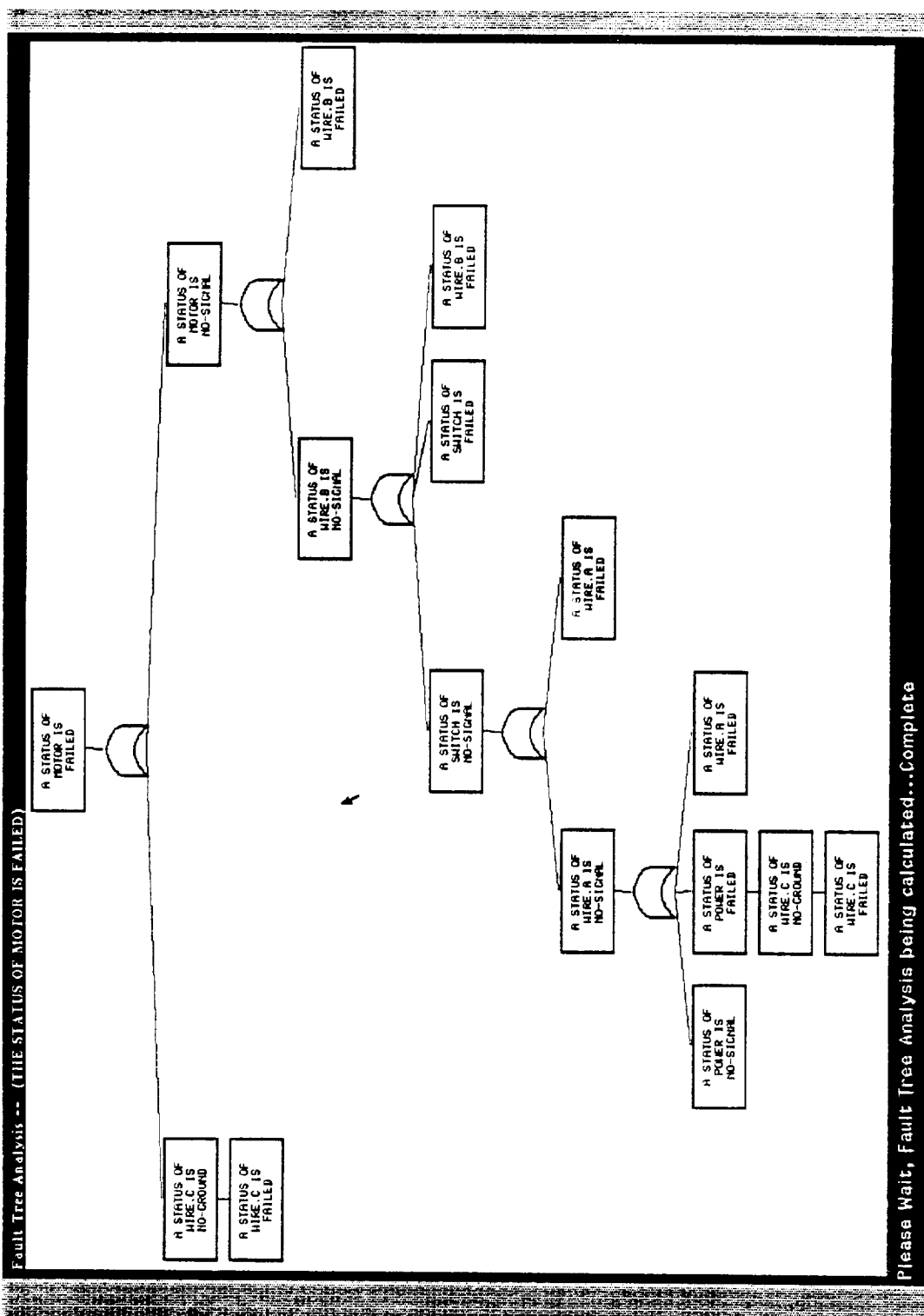


Figure 3.3-6. An FTA Analysis for the Demonstration Motor Control Model - Motor Failure

3.3.2.2 Library Objects

The rule capture mechanisms discussed in Section 3.3.2.1 require the user to define the behavior for each object found on an abstract diagram before the FMEAs and FTAs can be utilized. In many cases the objects are of a common type, such as wire, resistor or battery. Since the behavior of these objects is generally the same when the objects are of the same type, a library of objects has been defined which simplifies the specification of the behavioral rules. This library provides default behavior rules, which the user may use. These default rules considerably reduce the effort required of the user to specify the behavior of the many objects typically expected for an abstract diagram.

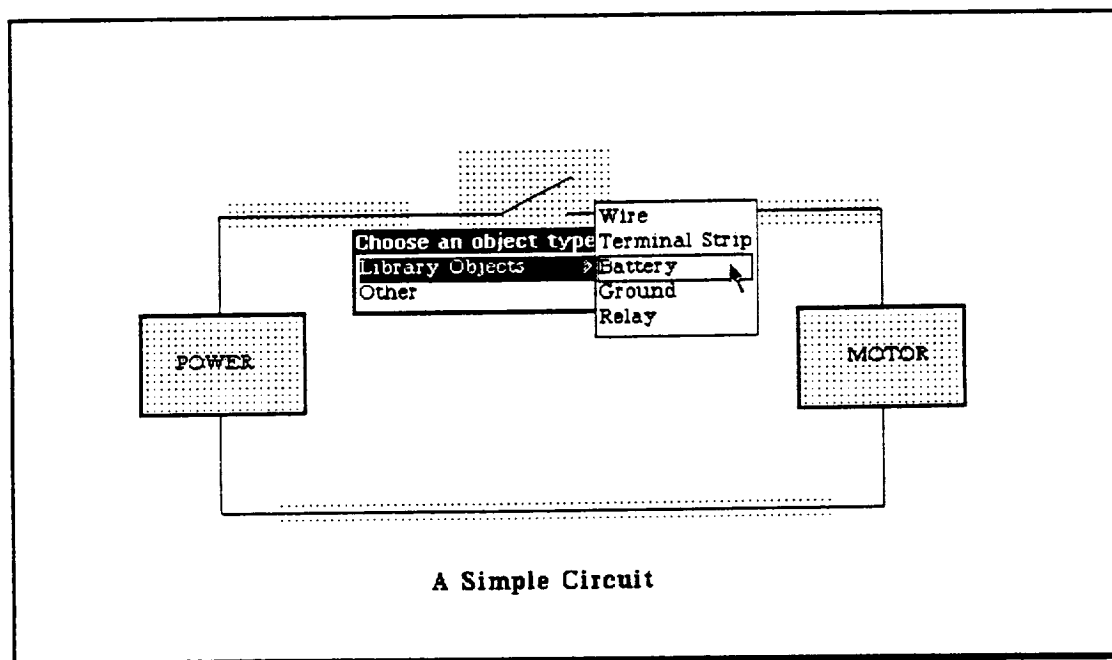


Figure 3.3-7. Menu of Object Type Choices - Used During Schematic Capture Process

Here, default behavior is defined for wires, batteries, terminal strips, relays and grounds. The following describes the default behavior assumed for wires in the library of objects. Three types of failures have been defined for wires: *no-signal*, *no-ground* and *failed*. The default behavior for a wire has been defined with two rules.

### Knowledge Capture Concepts

IF THE STATUS OF *WIRE* IS FAILED  
THEN THE STATUS OF *WIRE* IS NO-GROUND

---

IF THE STATUS OF *WIRE* IS FAILED  
THEN THE STATUS OF *WIRE* IS NO-SIGNAL

The default behavior assumes that if a wire undergoes a primary failure, then the wire will be incapable of either a proper grounding or of carry a current. Default behavior is defined in a similar fashion for the other library objects. This library of objects is one more design aid that enhances the users ability to develop FMEAs and FTA interactively. It is considered to be a significant knowledge capture concept.

#### **3.3.3 Other Options**

The KCS supplies a rich environment which supports the development of a vast array of useful rule-based models. The following is intended to briefly mention some of the many modeling possibilities.

The exploratory models provide discrete examples of rule-based models of selected FCS subsystems. The integration of many such rule-based models provides a viable approach to scale up this modeling capability. Such integration could implement a full scale FCS functionality, which might be driven by a user interface that displays the cockpit. Commands issued from this mouse sensitive display of cockpit aircraft control devices could be used to drive rule-based models. Individual subsystem performance displays would allow the user to browse the results of the cockpit commands. Component failure effects could be included to broaden the modeling and enable the user to explore degraded performance characteristics of the FCS.

Another rule-based modeling technique utilizes forward chaining to implement a synthesis paradigm. One application of this type of model could use the processing requirements associated with the S/W design to allocate S/W modules to rate groups. The allocation rules could consider the aircraft state, modes, and hardware status. Such a synthesis model could be used to study the computer loading as a function of these operational conditions.

### **3.4 The Supporting Technologies**

The KCS has not only been designed to integrate the design knowledge associated with the development of an FCS but also to recognize the necessity of working with the supporting technologies. One of the design guidelines has been to provide an open architecture which can be used to communicate with the supporting technologies. These supporting technologies include *computer-aided design* (CAD) and *computer-aided software engineering* (CASE) which are necessary to support the actual design of the hardware and software. In addition, these supporting technologies include *multimedia* which is necessary to enhance the communication with the user.

It is assumed that the sensor, effector, computer and communication hardware will be designed by CAD systems and that the software design will utilize CASE. Furthermore, it is assumed that these CAD/CASE systems can be used for other stages within the FCS life cycle. However, these individual CAD/CASE systems are viewed as being limited in their capabilities, namely that they only focus upon the needs of one (or perhaps a small number) of the many elements of the overall FCS. The KCS is designed to be compatible with these individual systems either on the same platform or over a network.

It is also assumed that when KCS is integrated with CAD and CASE systems, that these CAD/CASE capabilities will work closely with both the authoring and the browsing capabilities of the KCS. These systems will provide the properties and the values for the objects in the semantic network. In addition, inputs to these systems can also be derived from the KCS. Furthermore, the capabilities of these systems can be integrated into the rule-based models.

The KCS is also designed to be used with multimedia technology. The use of speech recognition for commands and audio/video devices for output offer natural ways with which to implement the user interface. Multimedia mechanisms can also be integrated with the KCS either in the same platform or over a network. Here again, the interface will utilize the object oriented nature of the KCS. The speech recognition mechanisms will manipulate the properties of the objects in the semantic network. The audio/visual outputs will reflect the appropriate output properties of the objects in the semantic network.

## Knowledge Capture Concepts

Once the KCS knowledge base, CAD devices and multimedia capabilities have been merged, it is possible to view this integration of capabilities as one which can mimic some of those capabilities projected by HAL (a fictional computer system) in the 1968 science fiction novel *2001: A Space Odyssey*. In this novel, a spaceship computer with human characteristics interacts with an astronaut using audio and visual communication. Among HAL's many implied capabilities, the computer provides intelligent information regarding the spaceship's control system.

In conjunction with the KCS, the speech recognition mechanisms can be used to augment or replace the mouse sensitive input displays used to browse the semantic network and interrogate the KCS regarding system status and performance. The audio/visual devices can augment the current KCS displays and provide appropriate responses. These projected capabilities are possible with today's technology. They are not science fiction.

The semantic network and the associated rule-based knowledge are assumed to contain the deeper knowledge which is necessary to integrate the many elements of the FCS. The KCS development has focused on the authoring, browsing and rule-based tools in this initial development phase. The integration of the CAD and multimedia mechanisms is assumed to be project dependent and to occur at a subsequent time. The open interface is intended to allow this integration to take place gracefully.

In summary, the CAD/CASE systems are recognized as being mandatory in the accomplishment of the designs in each of the individual disciplines. The KCS is viewed as being necessary to integrate the knowledge developed by these CAD/CASE systems. The capabilities of multimedia are viewed as being necessary to support the user interface. When these speech, audio and video capabilities are combined with the KCS, which has integrated the design knowledge, the combination can produce a remarkably intelligent apprentice for this sophisticated and complex problem domain.



## The Semantic Network

### 4.0 The Semantic Network

This project focused upon the development of a KCS. In the course of the development, a limited amount of design information for the HARV FCS has been captured. A primary purpose of this design capture effort was to exercise and illustrate the capabilities of the KCS. The design knowledge is captured in a semantic network, which integrates the design knowledge from four design disciplines: system, H/W, S/W and utilities. The design knowledge from each discipline is captured in individual realms of knowledge. Each realm is comprised of a highly correlated set of hierarchical data structures. This section describes the selected design knowledge which has been captured in the individual design realms which comprise the integrated semantic network.

Figure 4-1 depicts the user interface at the top level of the KCS which gives the user access to the four realms which comprise the semantic network. The individual diagrams in each realm possess their own capability to browse the multi-realm semantic network.

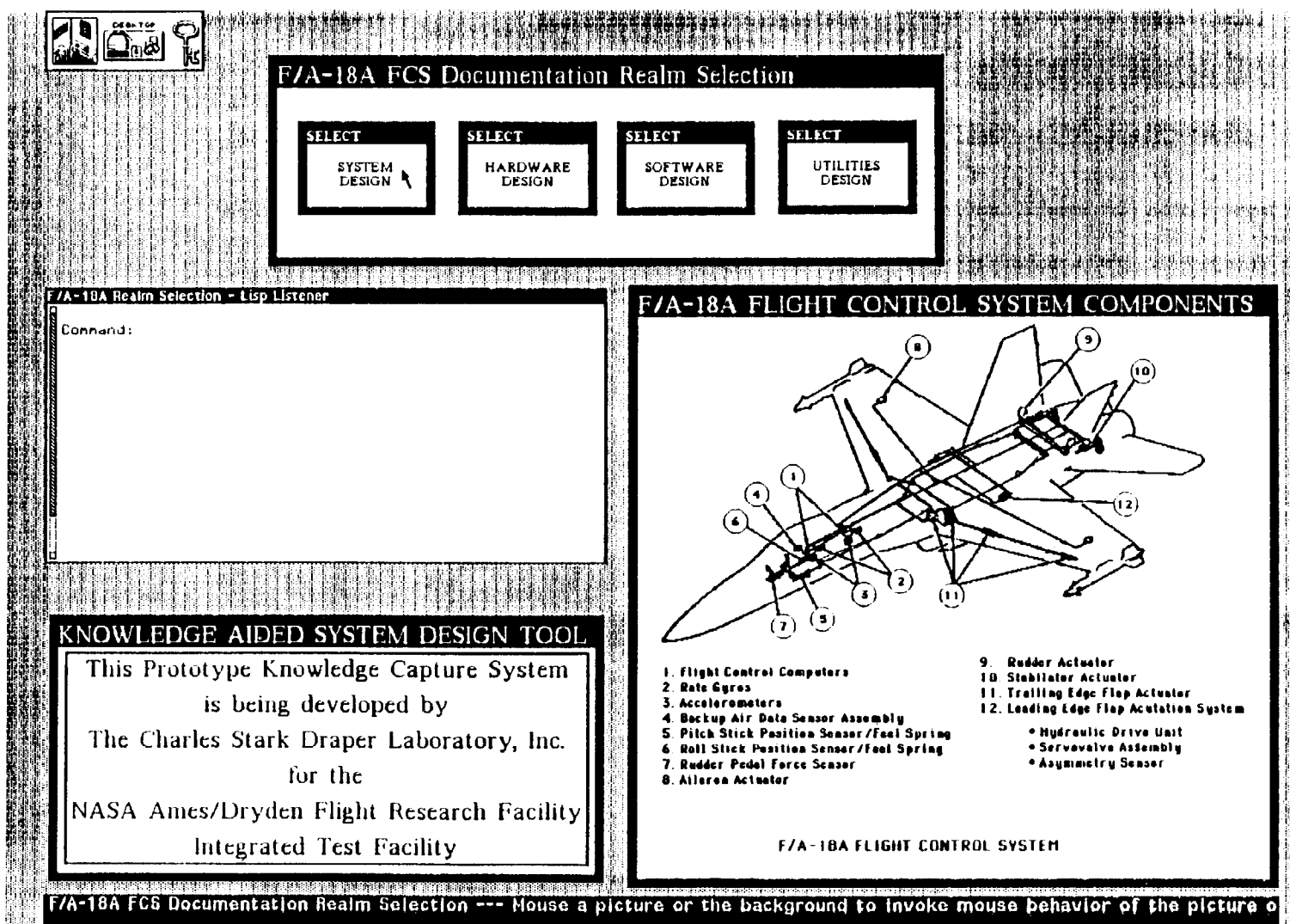


Figure 4-1. F/A-18A Top Level Realm Selection User Interface

#### 4.1 The System Design Realm

The following goals guided the selection of the FCS design which was captured in the system design realm.

- Demonstrate the nature of system design knowledge
- Demonstrate the synergistic combination of semantic networks and SA decomposition
- Demonstrate the role of authoring and browsing for design knowledge captured within the constraints of the SA methodology
- Demonstrate the use of behavioral models

The top level DFD, which is depicted in Figure 4.1-1, identifies the three major aspects of the system design for the FCS:

1. System Management and Control
2. Flight Control
3. Actuator Management

Two of these aspects of the FCS system design have been decomposed, the flight control and the actuator management. The flight control characteristics have been expanded to indicate the functionality in level 3 DFDs (a 3X expansion). These DFDs are depicted in Figures 4.1-2 through 4.1-6. This decomposition reaches deeply enough into the FCS system design to illustrate the nature of a top down decomposition and the nature of the resulting semantic network.

In addition, the actuator management characteristics have been decomposed (expanded) to indicate the functionality of the spin chute emergency system. The spin chute deployment system is defined in greater detail in the H/W design realm. This HARV system design decomposition depicts an overview of the actuator management in a level 1 DFD (a 1X expansion) and the spin chute emergency system functionality in a level 2 DFD (a 2X expansion). These DFDs are depicted in Figures 4.1-7 and 4.1-8.

The semantic network hierarchies for the process, data flow and external objects defined by this decomposition are depicted in Figures 4.1-9 through 4.1-11. These hierarchies form a

### The Semantic Network

parent/child relationship with inheritance features. The linkage between the objects, which are in separate networks, is defined by properties of the individual objects (e.g. source, destination, inputs and outputs).

The illustrative dynamic behavioral models for the nose wheel steering (NWS) and spin chute deployment systems are linked to the appropriate processes defined by this system design decomposition. These behavioral models are described in Section 5.0.

As a user interface, the system design DFDs provide a hierarchical access to these models. Starting with the level 0 flight control process, it is possible to mouse down through three process expansions and to then select the NWS behavioral model. The NWS behavioral model indicates the behavior of an FCS subsystem as a function of aircraft modes. Similarly, it is possible to browse downward through the actuator management process hierarchy and reach the spin chute emergency system. Here, FMEA and FTA models can be accessed from the process menu for the spin chute emergency system which enable the user to explore the effects of component failures.

F/A-18A FCS TOP LEVEL DFD

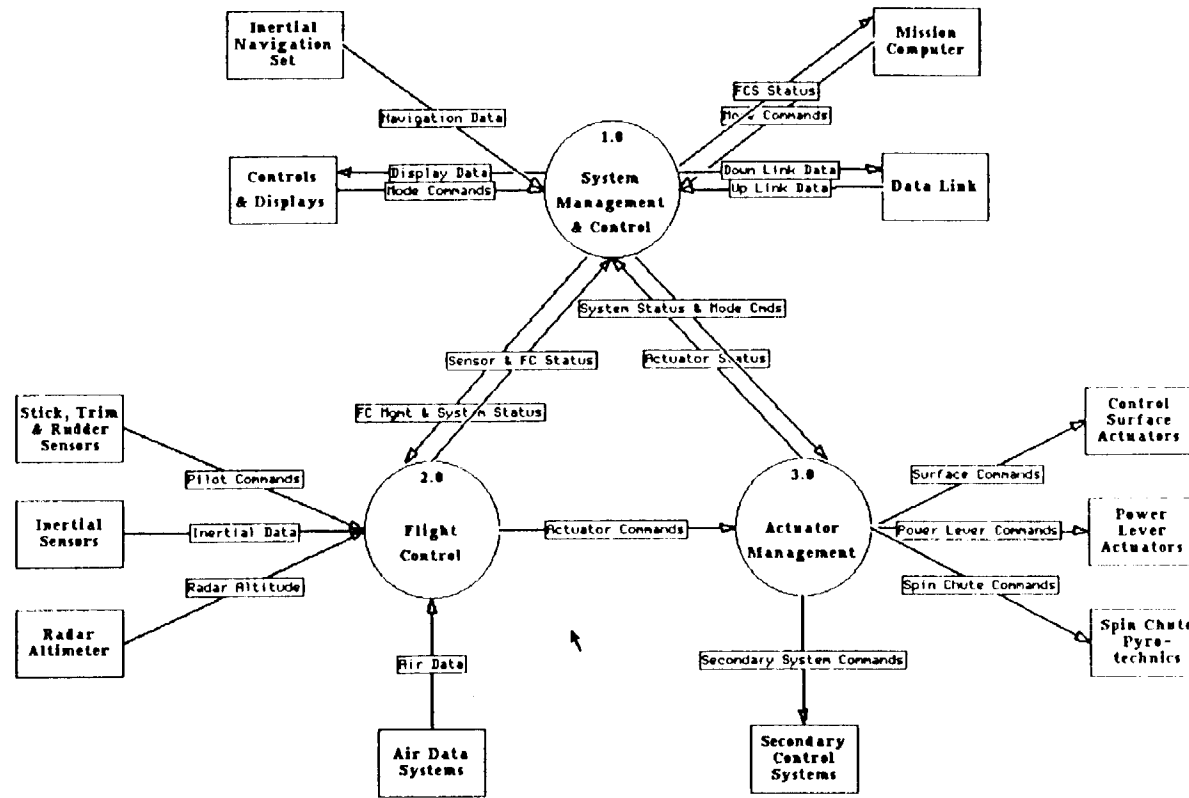


Figure 4.1-1. F/A-18A System Design Level 0 DFD - The Top Level Data Flow Diagram

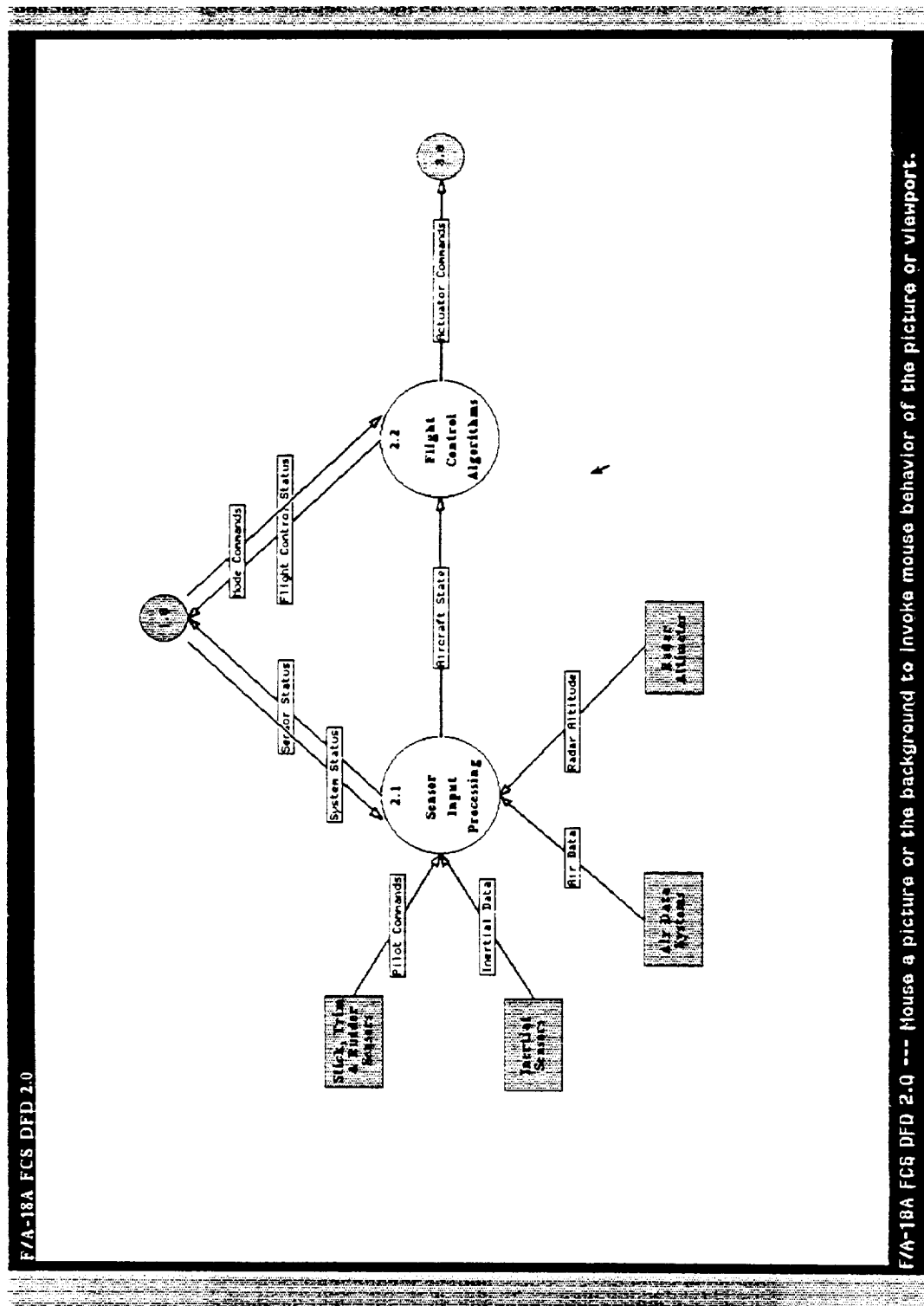
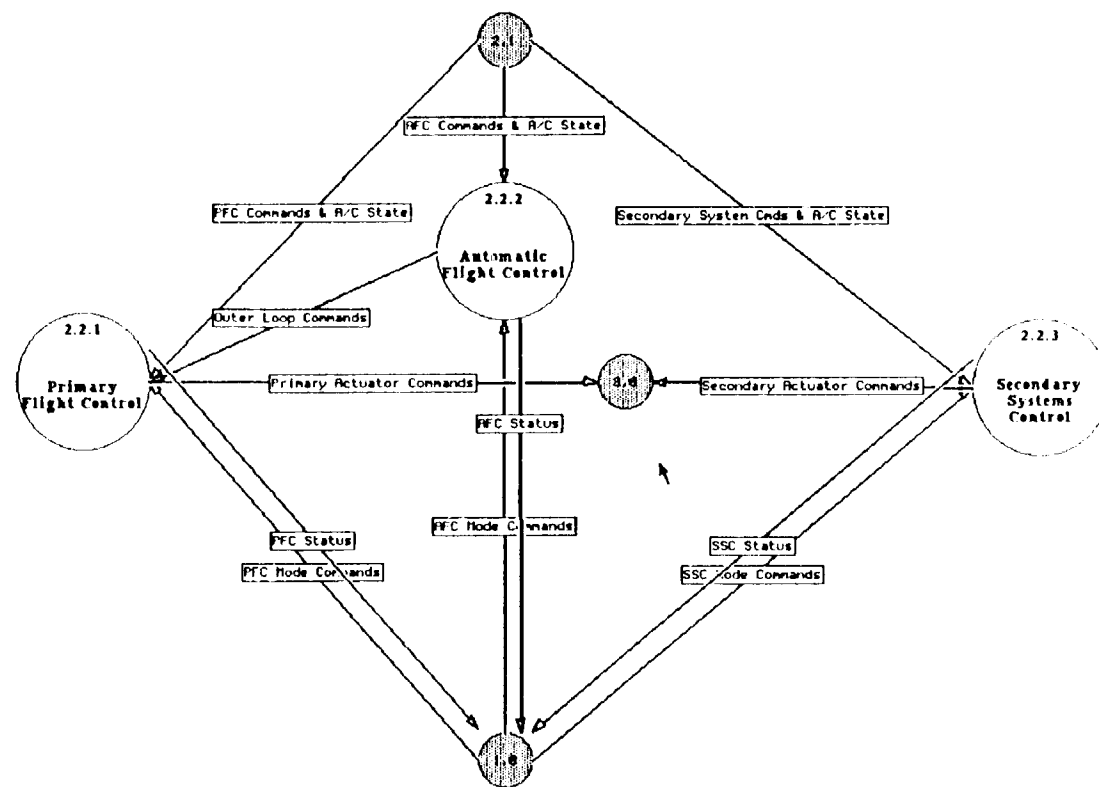


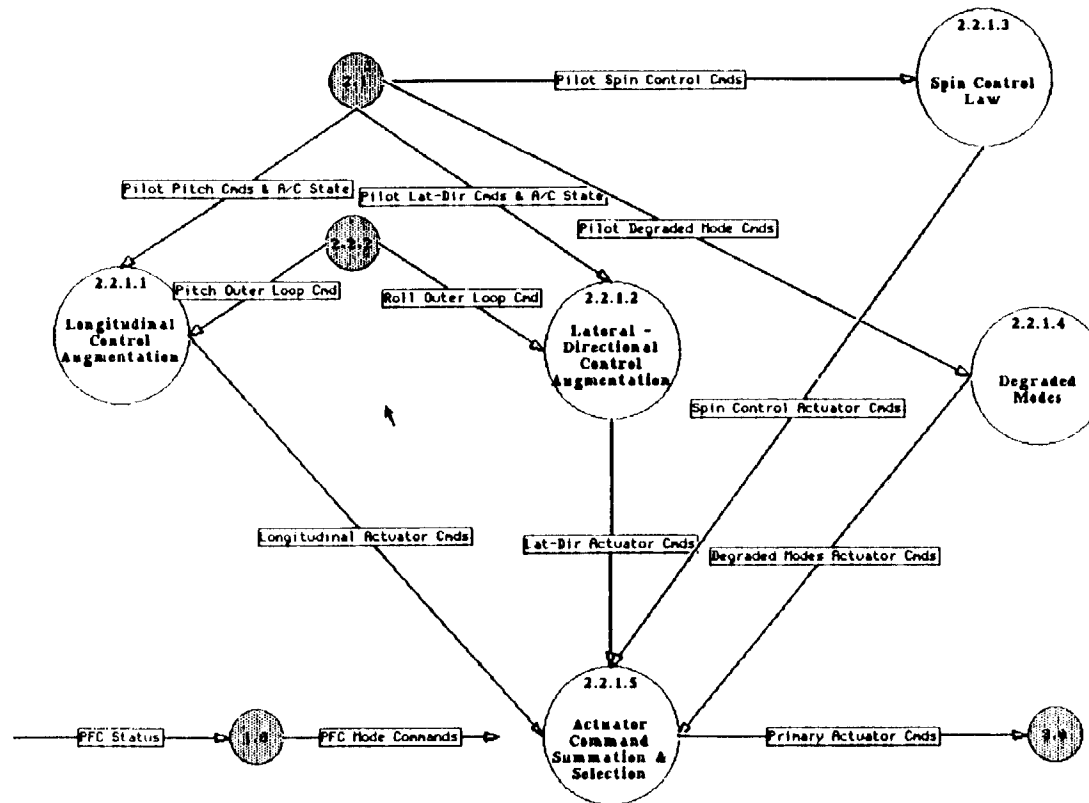
Figure 4.1-2. F/A-18A System Design Level 1 DFD: DFD 2.0 - Flight Control



F/A-18A FCS DFD 2.2 --- Mouse a picture or the background to invoke mouse behavior of the picture or viewport.

Figure 4.1-3. F/A-18A System Design Level 2 DFD: DFD 2.2 - Flight Control Algorithms

F/A-18A FCS DFD 2.2.1

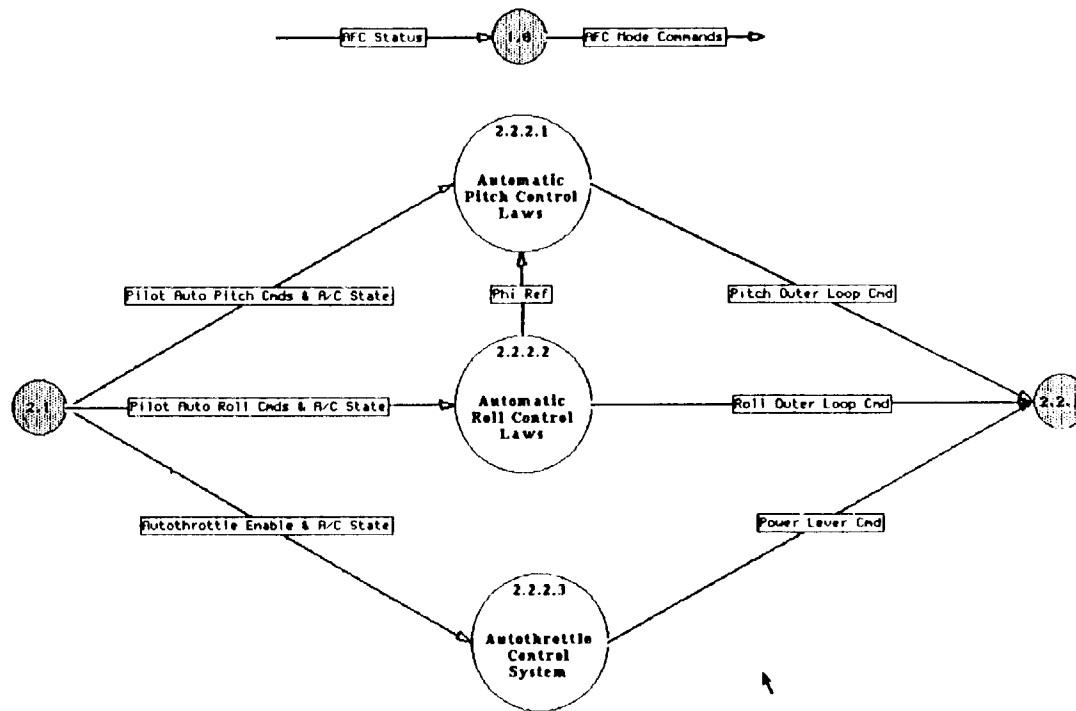


F/A-18A FCS DFD 2.2.1 --- Mouse a picture or the background to invoke mouse behavior of the picture or viewport.

Figure 4.1-4. F/A-18A System Design Level 3 DFD: DFD 2.2.1 - Primary Flight Control



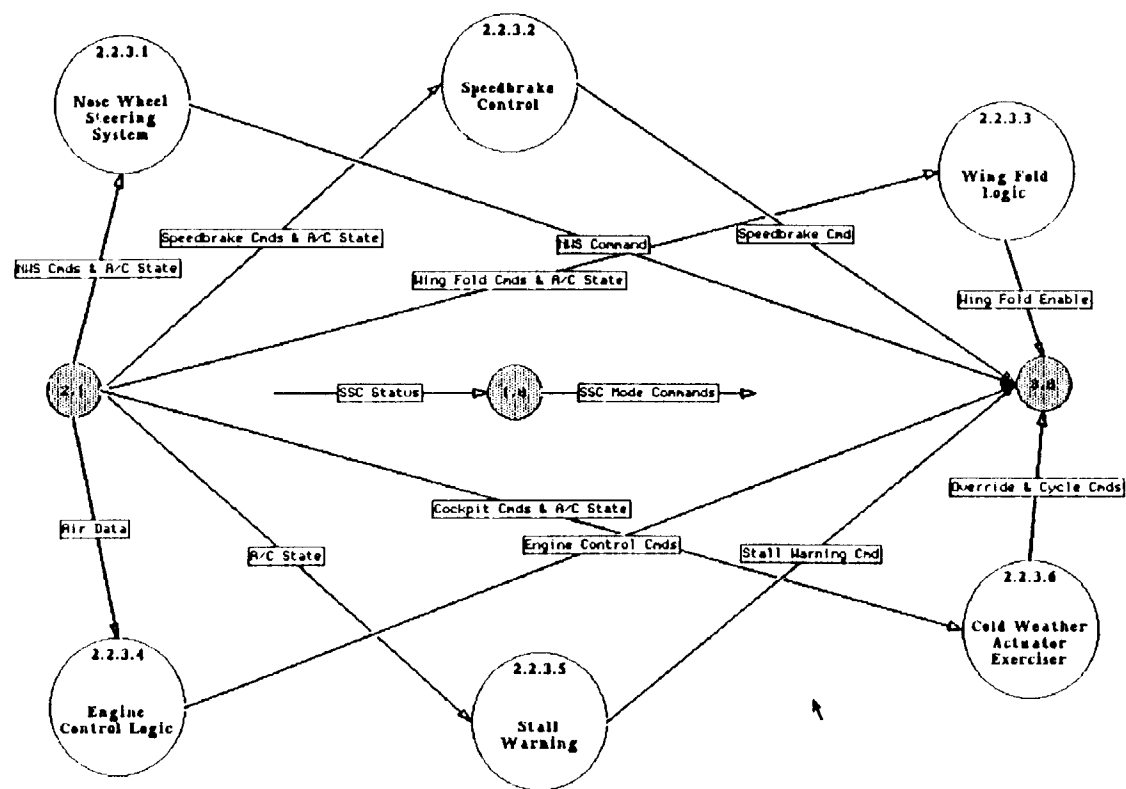
F/A-18A FCS DFD 2.2.2



F/A-18A FCS DFD 2.2.2 --- Mouse a picture or the background to invoke mouse behavior of the picture or viewport.

Figure 4.1-5. F/A-18A System Design Level 3 DFD: DFD 2.2.2 - Automatic Flight Control

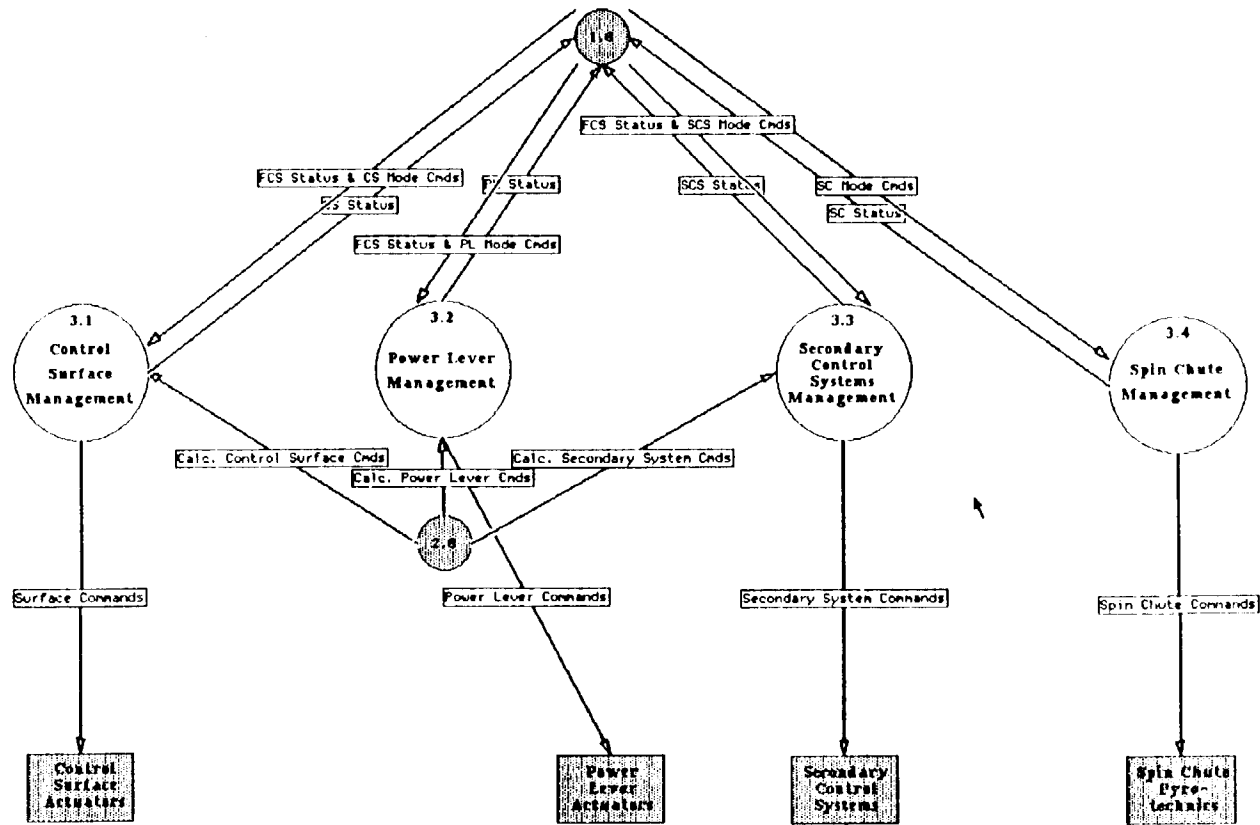
F/A-18A FCS DFD 2.2.3



F/A-18A FCS DFD 2.2.3 --- Mouse a picture or the background to invoke mouse behavior of the picture or viewport.

Figure 4.1-6. F/A-18A System Design Level 3 DFD: DFD 2.2.3 - Secondary Systems Control

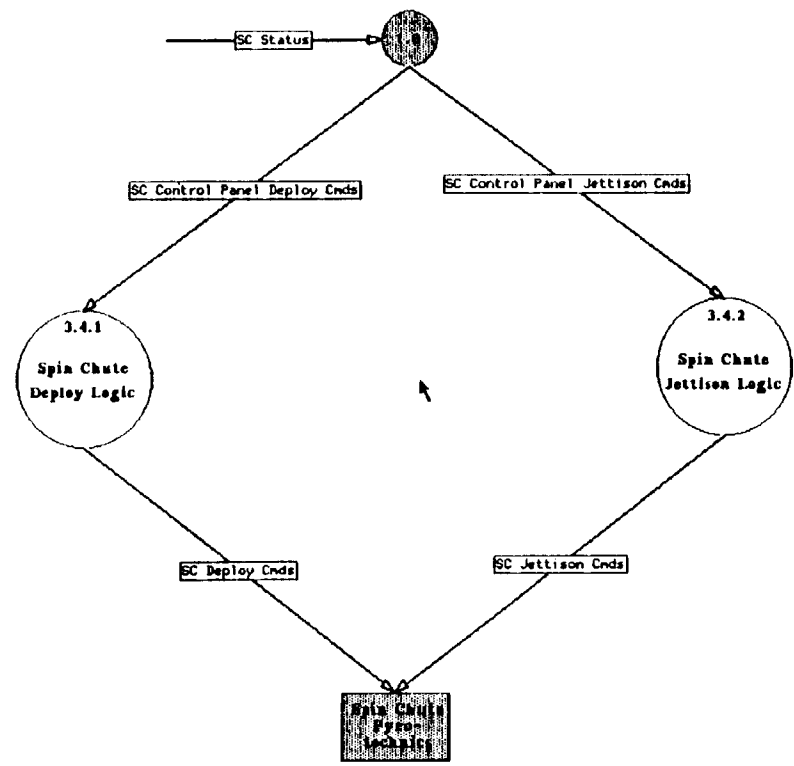
F/A-18A FCS DFD 3.0



F/A-18A FCS DFD 3.0 --- Mouse a picture or the background to invoke mouse behavior of the picture or viewport.

Figure 4.1-7. F/A-18A System Design Level 1 DFD: DFD 3.0 - Actuator Management

F/A-18A FCS DFD 3.4



F/A-18A FCS DFD 3.4 --- Mouse a picture or the background to invoke mouse behavior of the picture or viewport.

Figure 4.1-8. F/A-18A System Design Level 2 DFD: DFD 3.4 - Spin Chute Management

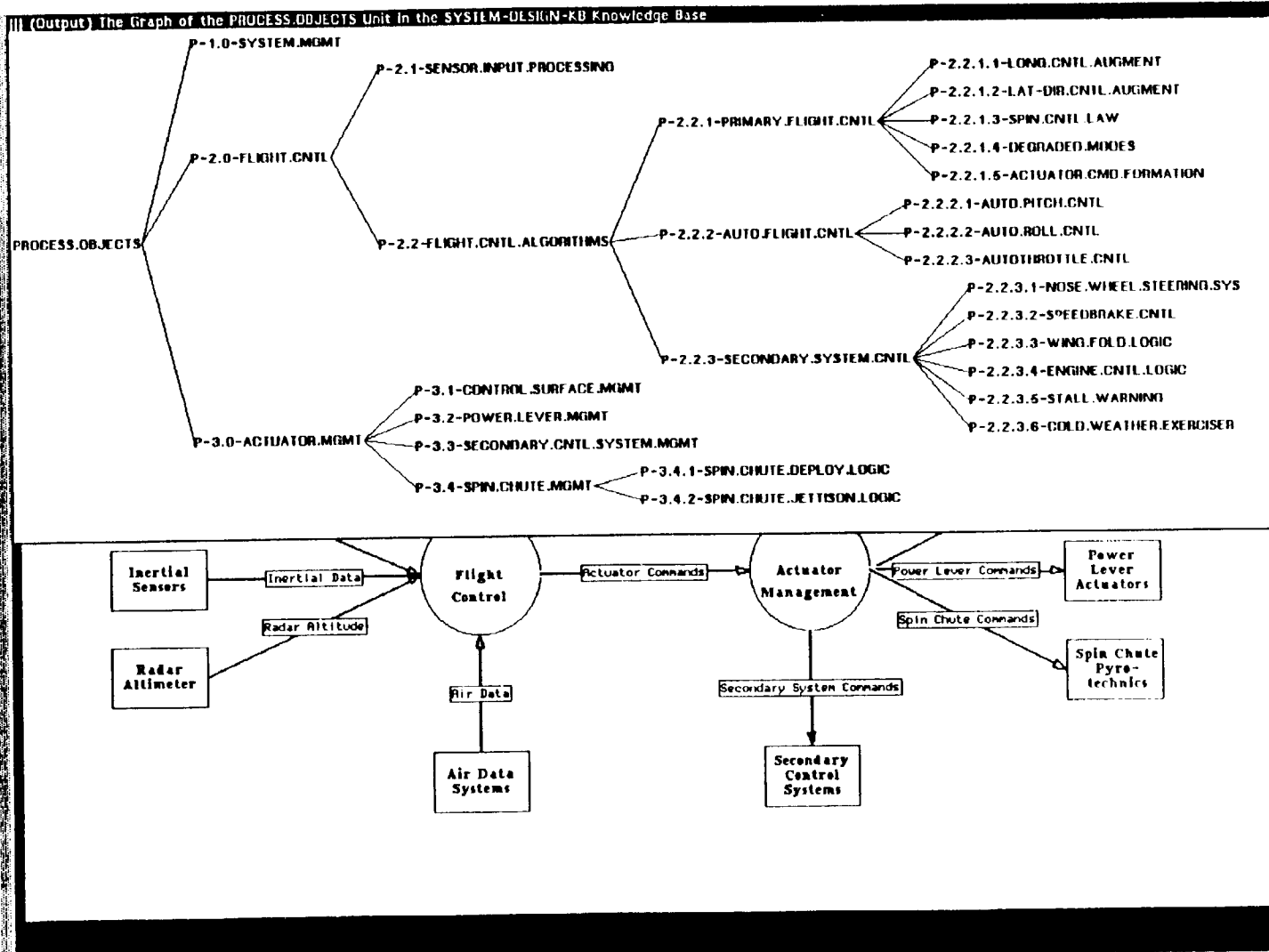


Figure 4.1-9. F/A-18A System Design Hierarchy of Process Objects

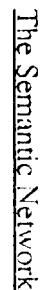


Figure 4.1-10. F/A-18A System Design Hierarchy of Data Flow Objects

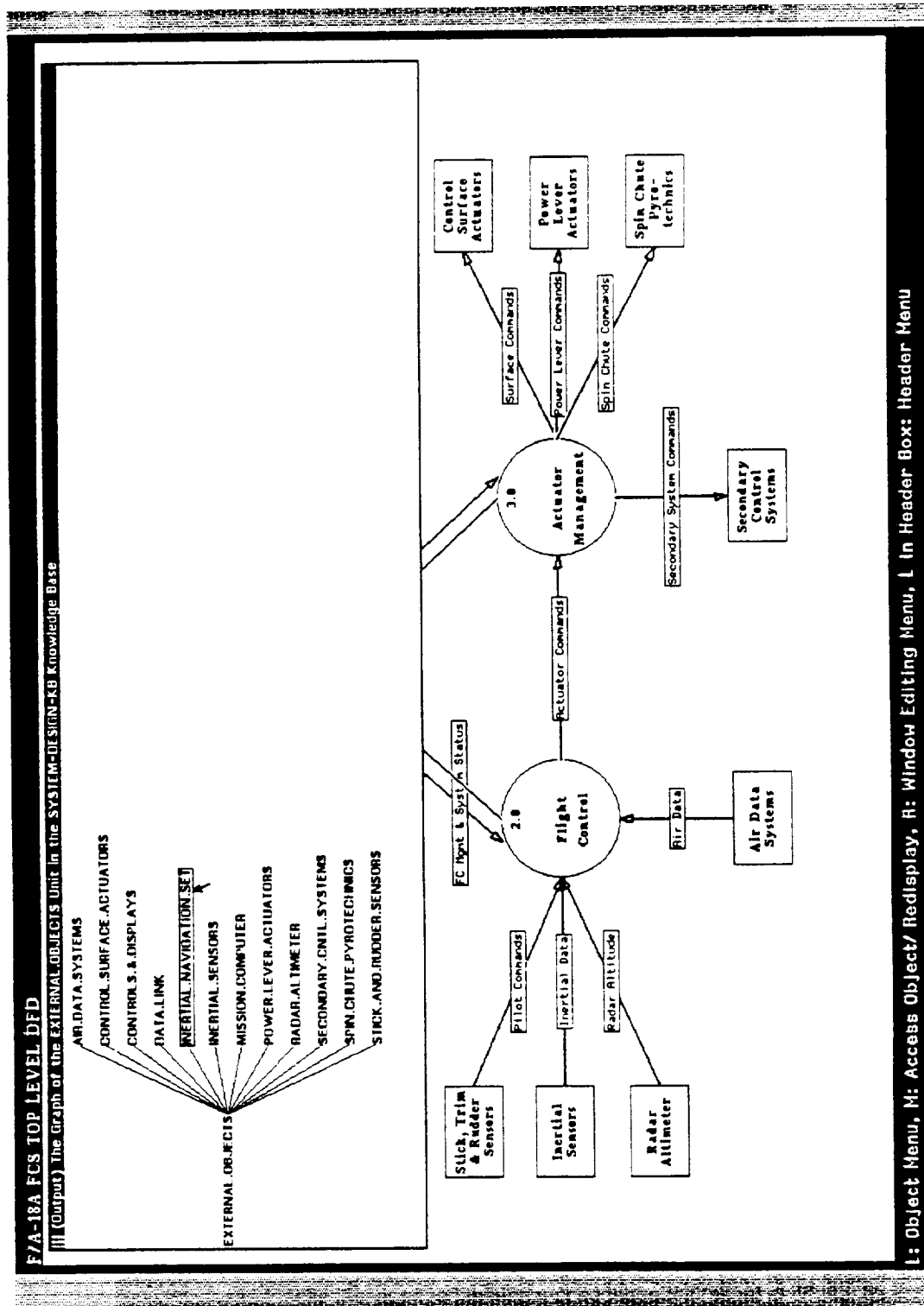


Figure 4.1-11. F/A-18A System Design Hierarchy of External Objects

## The Semantic Network

### 4.2 The H/W Design Realm

The following goals guided the selection of the FCS design which was captured in the H/W design realm.

- Demonstrate the nature of H/W design knowledge
- Demonstrate the synergistic combination of semantic networks and H/W decomposition
- Demonstrate the role of authoring and browsing for design knowledge captured within the constraints of the H/W methodology
- Demonstrate the use of generic component properties
- Demonstrate the use of scanned schematics and their linkage to the semantic network objects
- Demonstrate behavioral failure models
- Demonstrate the use of H/W and S/F objects in rule-based models

Figure 4.2-1 depicts a conventional abstraction for a top level H/W diagram with sensors, computers, effectors and the information flow. Figures 4.2-2 through 4.2-6 depict a representative set of the level 1 H/W diagrams. These figures depict level 1 H/W diagrams for

- the rate sensors,
- flight control computer A
- the left stabilator actuator,
- the left trailing edge flap actuator and
- the left aileron actuator.

The rate sensor H/W diagram defines a block diagram decomposition which indicates the redundant rate sensor design. Figure 4.2-7 depicts the hierarchy and functional properties for the channel 1 roll rate gyro. The upper hierarchy display window indicates the dual inheritance path (H/W.OBJECTS & ROLL.RATE.SENSOR.GENERIC.PROPERTIES) for this particular roll rate gyro. The lower property display window indicates the roll rate gyro properties which were inherited from the ROLL.RATE.SENSOR.GENERIC.PROPERTIES template. The templates for the pitch rate and yaw rate gyro functional properties are



## The Semantic Network

similar to the one used for the roll rate gyros. This H/W diagram demonstrates the nature of H/W decomposition and the use of generic problem domain properties.

Figures 4.2-3 through 4.2-6 illustrate the use of mechanical drawings in the H/W diagrams. Here, several representative diagrams indicate the use of this graphical representation in the decomposition of a H/W design. These mechanical drawings define the decomposition, when a user expands the top level H/W object associated with each of these H/W diagrams. The decomposition process for the H/W objects associated with the H/W diagrams depicted in Figures 4.2-3 through 4.2-6 are incomplete. Completion of the decomposition entails the use of the authoring mechanisms to define the H/W and S/F objects associated with the hierarchical decomposition process. In addition, it is necessary to define the linkage between the box/line representation and the mechanical drawing representation.

Figures 4.2-1 and Figures 5-5 through 5-13 depict a multi-level decomposition of a typical top level H/W object (the Spin Chute Pyrotechnics System). This system is decomposed from a top level abstraction, thru a mid-level definition and down to the component level. This decomposition starts with a box and line representation in the level 0 (Figure 4.2-1) and level 1 (Figure 5-7) H/W diagrams. Electrical drawings are introduced in the level 2 H/W diagrams and continue to be used as the spin chute deployment system is further decomposed (Figures 5-8 through 5-13). This particular decomposition illustrates the use of hotspots to correlate the drawing with the box/line representations. In addition, this system decomposition illustrates a typical coupling between the objects in the semantic network and the use of the inference engine. A detailed description of this typical use of the inference engine to develop behavioral models is given in Section 5.2

The semantic network hierarchies for the H/W objects and the S/F objects defined by this decomposition in the H/W design realm are depicted in Figures 4.2-8 and 4.2-9. Although these hierarchies represent only a small percentage of the HARV FCS, they can be seen to exceed the size of the display screen. In its final form, these hierarchies would be significantly larger and consist of hundreds (or probably thousands) of objects. The scrollable displays depicted in Figures 4.2-8 and 4.2-9 enable the user to explore these large hierarchies. These hierarchies form a parent/child relationship with inheritance features. The linkage between the objects, which are in separate networks, is defined by properties of the individual objects (e.g. source, destination, inputs and outputs).

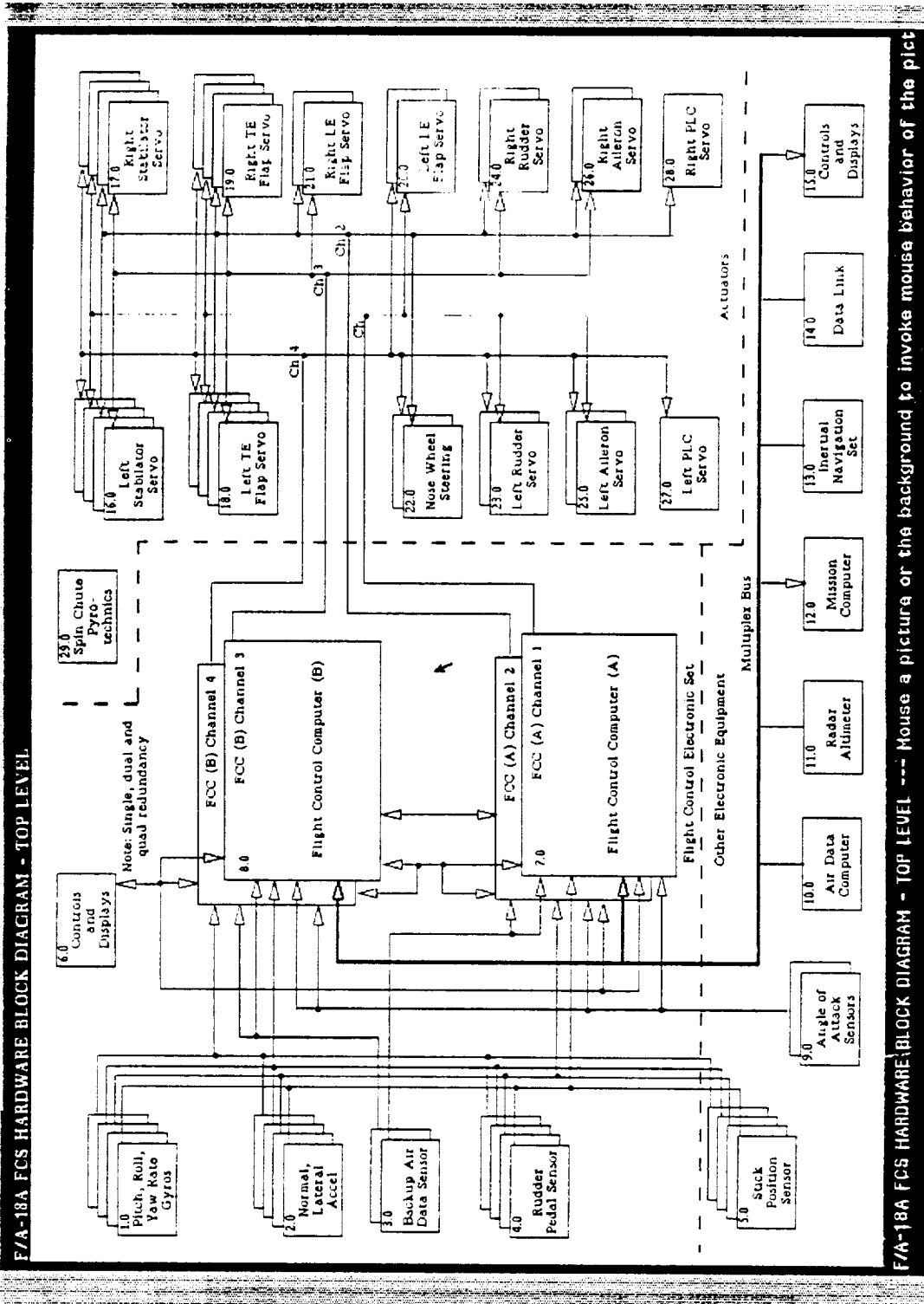
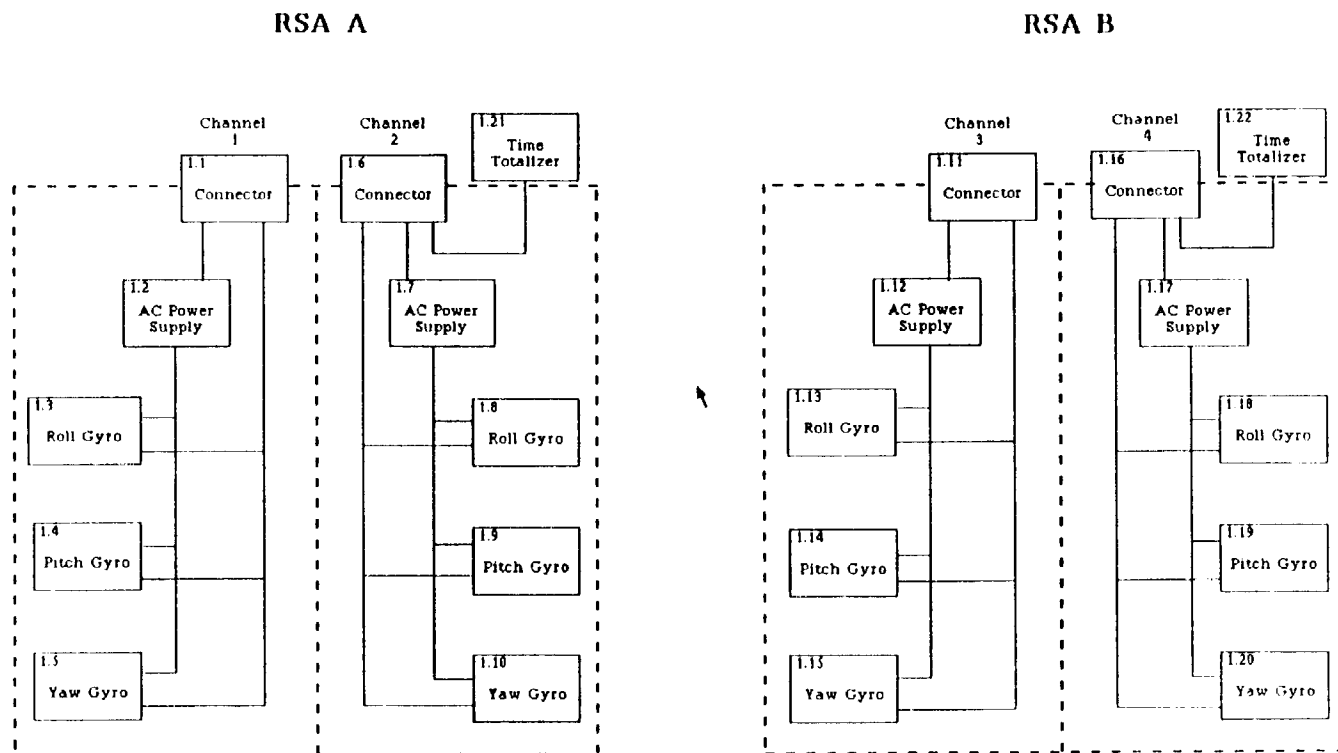


Figure 4.2-1. The Top Level H/W Diagram

F/A-18A FCS H/W OBJECT 1.0

Pitch, Roll, Yaw Rate Gyros



RATE SENSOR ASSEMBLY FUNCTIONAL DIAGRAM

F/A-18A FCS H/W OBJECT 1.0

Pitch, Roll, Yaw Rate Gyros --- Mouse a picture or the background to invoke mouse

Figure 4.2-2. H/W Diagram 1.0 - Pitch, Roll, Yaw Rate Gyros

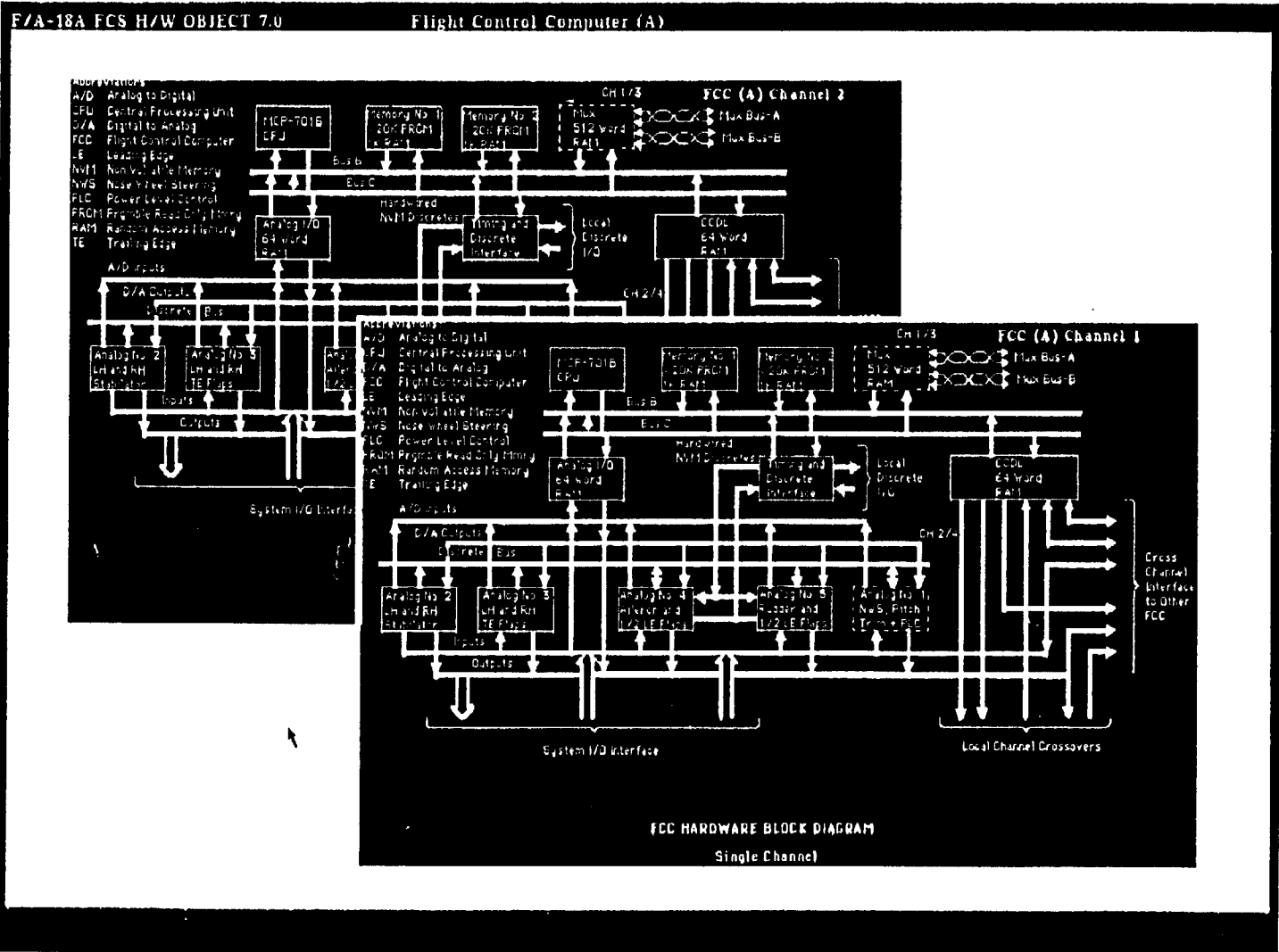
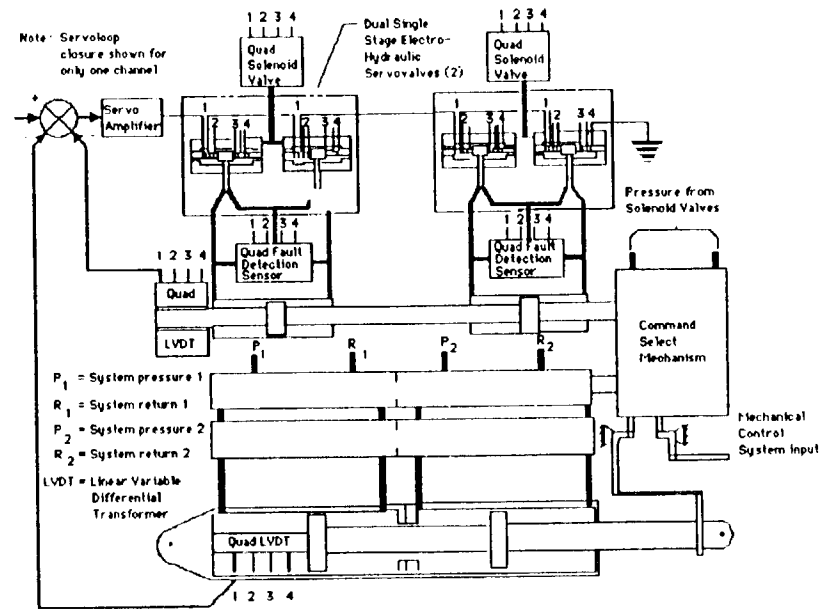


Figure 4.2-3. H/W Diagram 7.0 - Flight Control Computer (A) Channels 1 & 2

F/A-18A FCS H/W OBJECT 16.0

Left Stabilator Servo



STABILATOR ACTUATOR SCHEMATIC

F/A-18A FCS H/W OBJECT 16.0

Left Stabilator Servo --- Mouse a picture or the background to invoke mouse behav

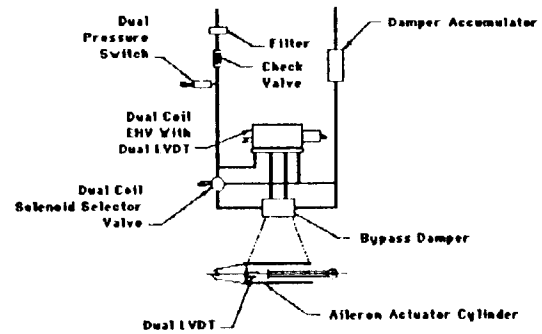
Figure 4.2-4. H/W Diagram 16.0 - Left Stabilator Servo



**Figure 4.2-5. H/W Diagram 18.0 - Left TE Flap Servo**

F/A-18A FCS H/W OBJECT 25.0

Left Aileron Servo



AILERON ACTUATOR SCHEMATIC

F/A-18A FCS H/W OBJECT 25.0

Left Aileron Servo --- Mouse a picture or the background to invoke mouse behavior

Figure 4.2-6. H/W Diagram 25.0 - Left Aileron Servo

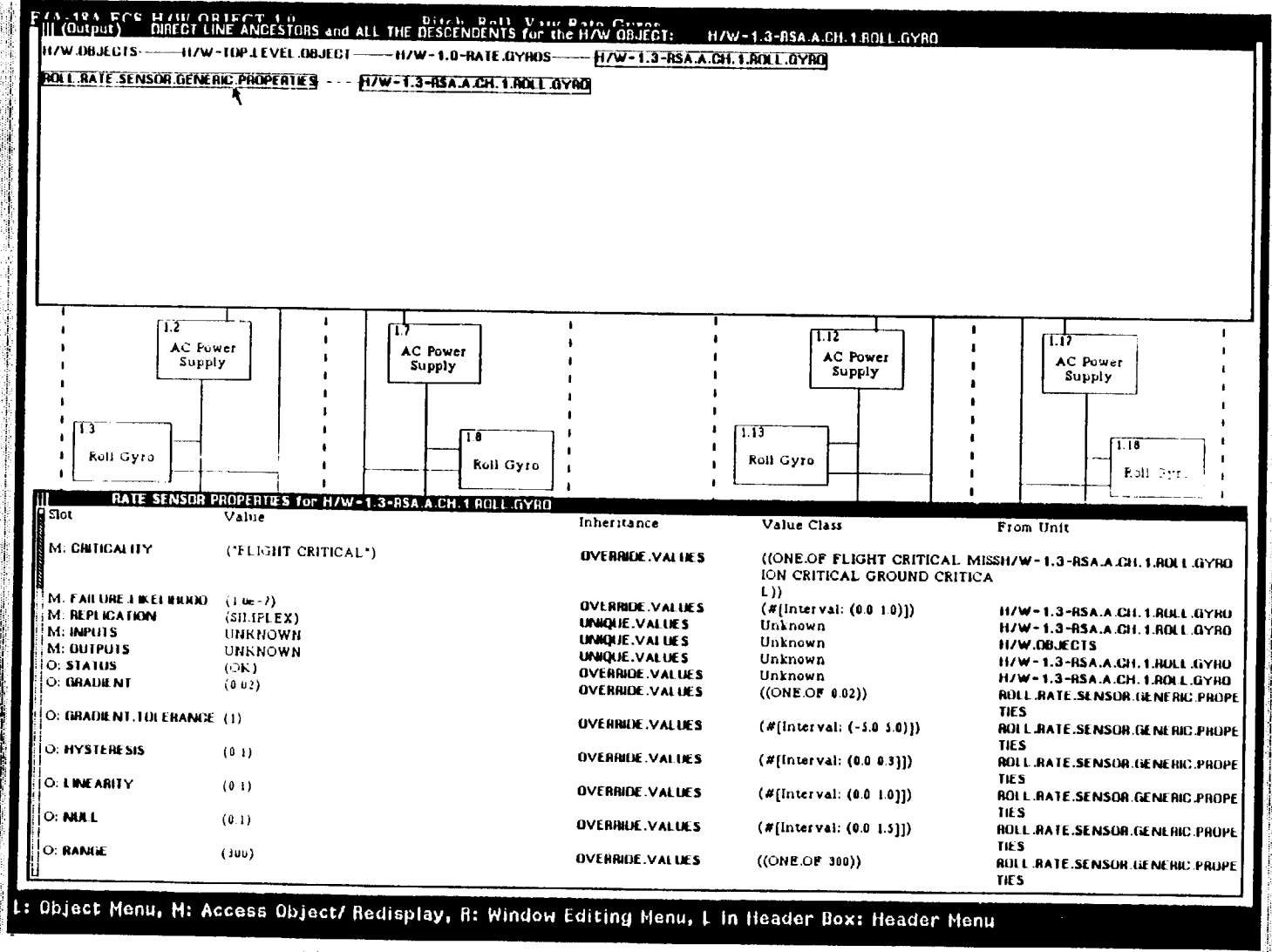


Figure 4.2-7. Roll Rate Sensor Functional Properties Template



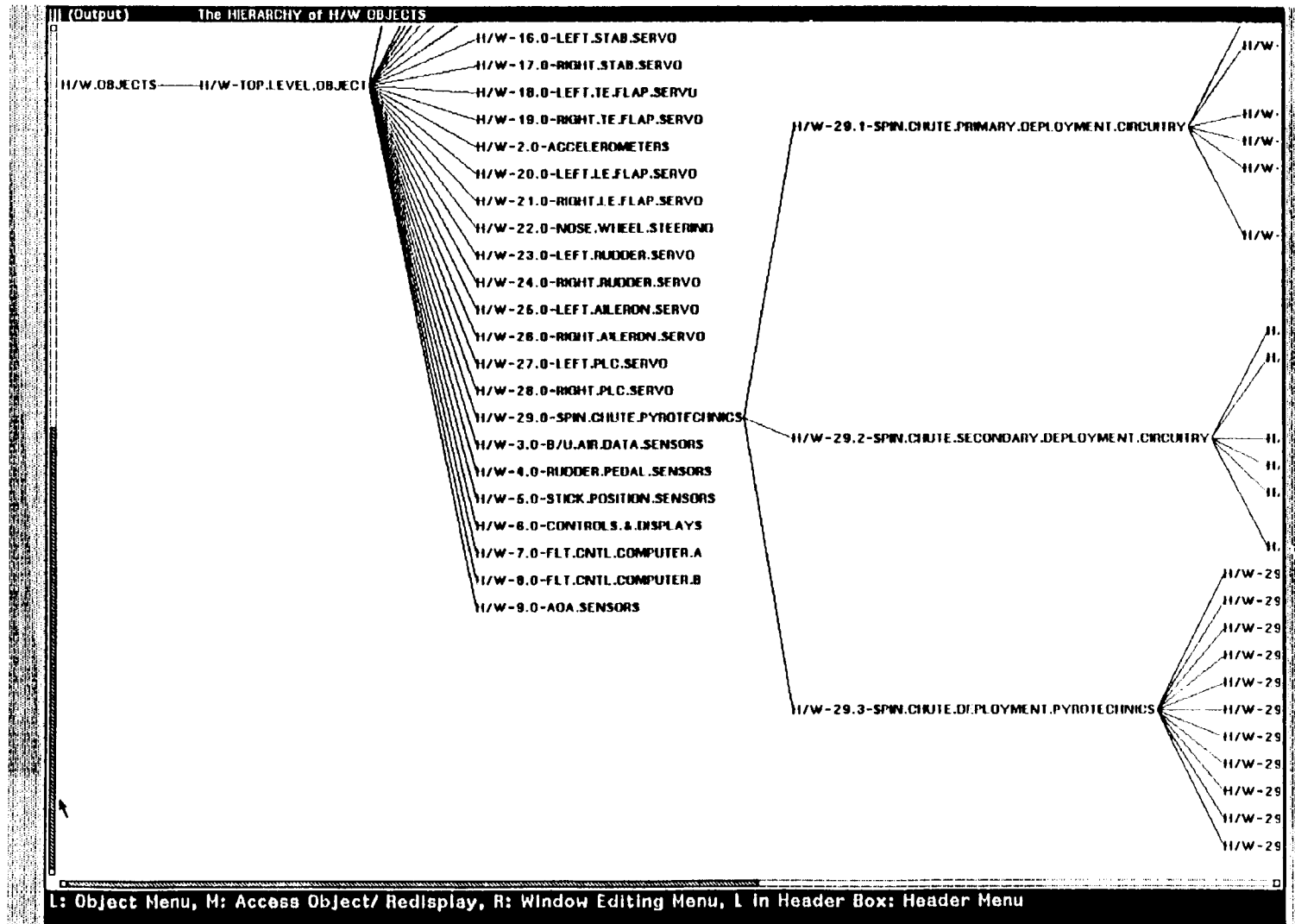


Figure 4.2-8. F/A-18A H/W Design Hierarchy of H/W Objects

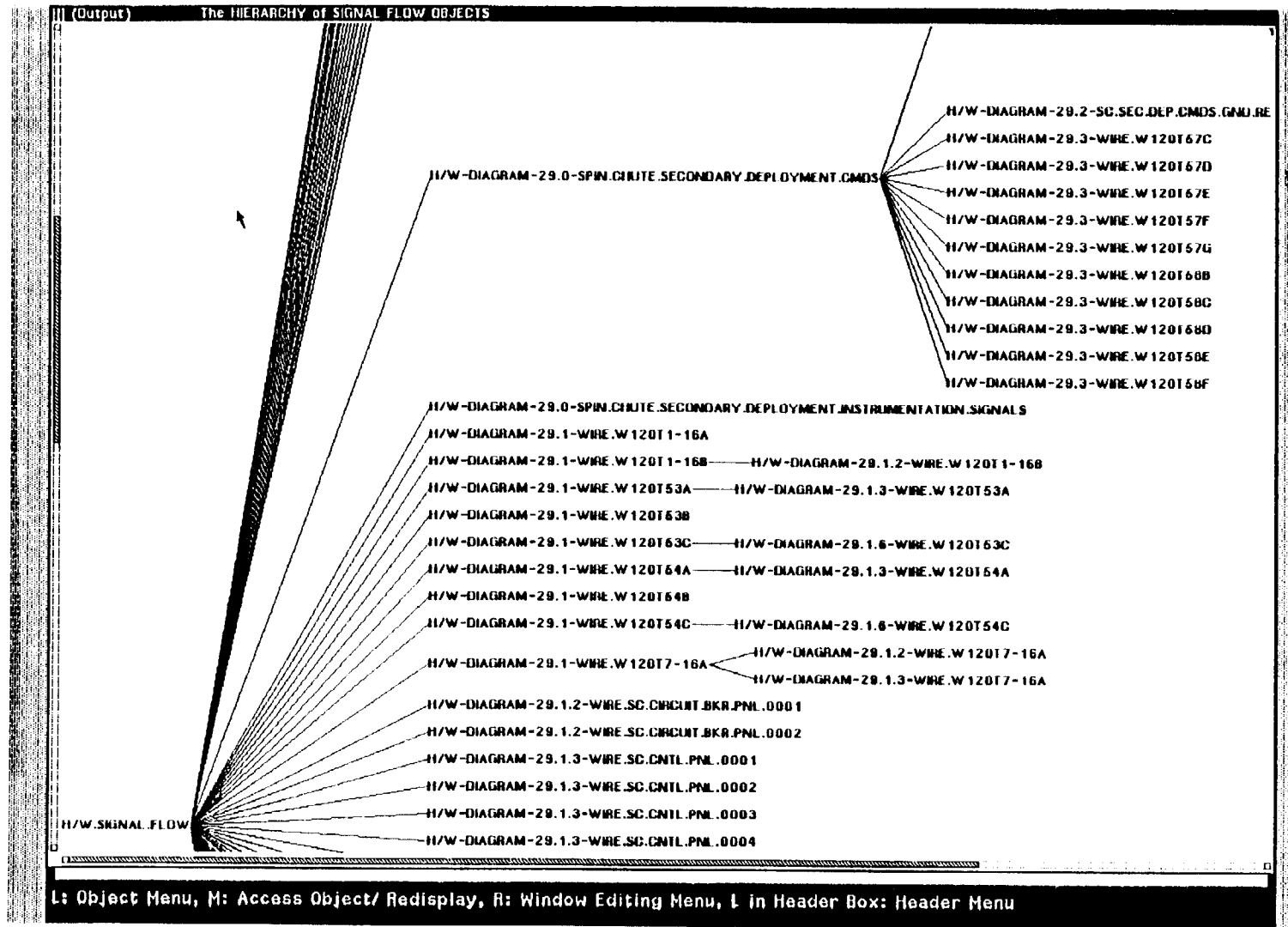


Figure 4.2-9. F/A-18A H/W Design Hierarchy of Signal Flow Objects

## The Semantic Network

### 4.3 The S/W Design Realm

The following goal guided the selection of the FCS design which was captured in the S/W design realm.

- Demonstrate the nature of S/W design knowledge

The top level DFD, which is depicted in Figure 4.3-1, identifies the major S/W processes for the FCS:

1. The Executive Module
2. The Flight Control Modules
3. The Built-In Test Modules
4. The Mission Compute Data Management Module

Two of the FCS S/W design processes have been decomposed, the flight control modules (Figure 4.3-2) and the built-in test modules (Figure 4.3-3). This decomposition indicates enough of the FCS S/W design to illustrate the nature of a S/W design top down decomposition and the nature of the resulting semantic network.

The semantic network hierarchies for the process, data flow and external objects defined by this decomposition are depicted in Figures 4.3-4 through 4.3-6. These hierarchies form a parent/child relationship with inheritance features. The linkage between the objects, which are in separate networks, is defined by properties of the individual objects (e.g. source, destination, inputs and outputs).

It should be recognized that the discussion in this report is focused on the development of the KCS. Here, this focus excludes the CASE systems which are mandatory to support the efforts associated with automated system S/W. In a future full scale application of the KCS, it is assumed that network access to these CASE tools would be utilized. One simple example of this form of tool integration would allow the user to display the code associated with a particular process. In a typical S/W development project, a CASE tool which supports configuration control, would be utilized. Access to the CASE configuration

### The Semantic Network

control display mechanism by the KCS would enable the KCS user to browse the code associated with objects in the KCS S/W hierarchies in the semantic network.

F/A-18A - S/W DESIGN - TOP LEVEL DIAGRAM

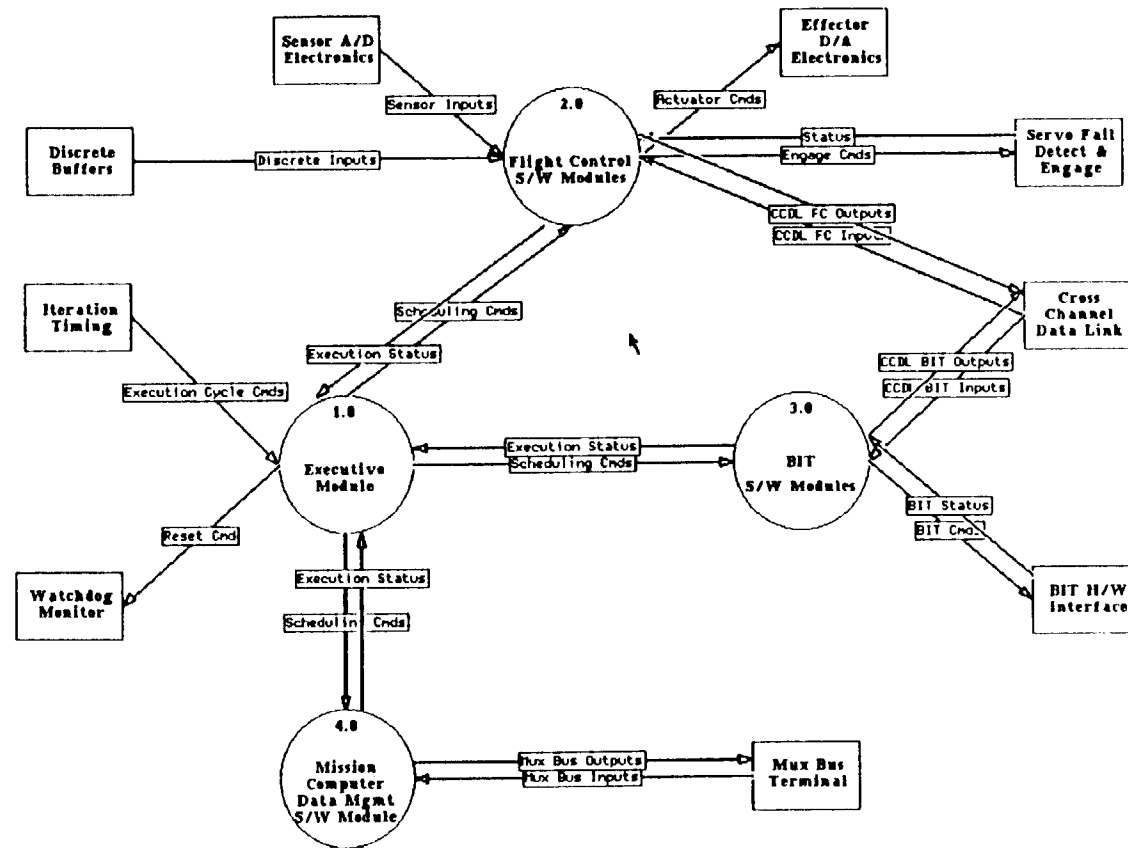


Figure 4.3-1. F/A-18A S/W Design Level 0 DFD - The Top Level Data Flow Diagram

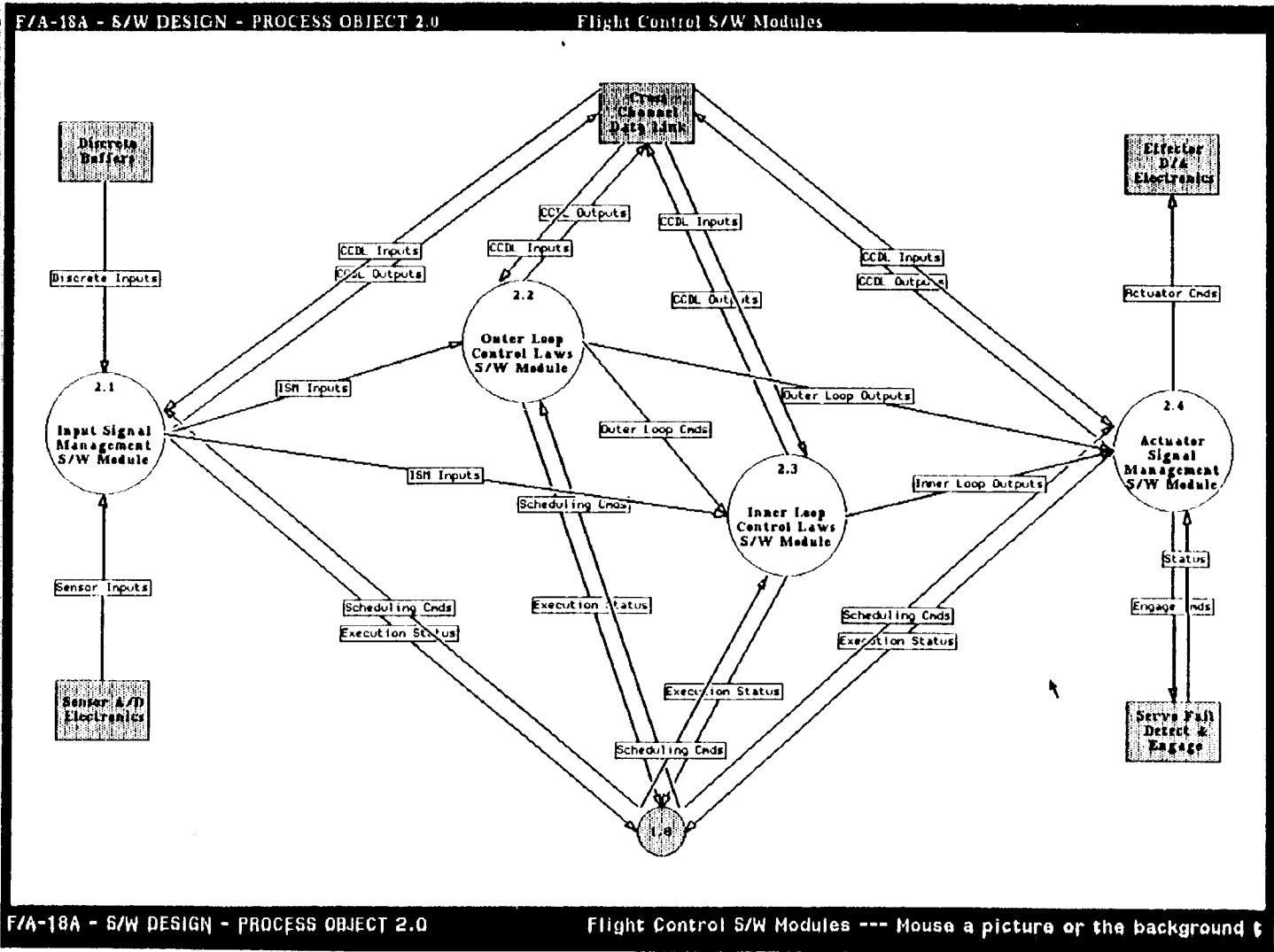
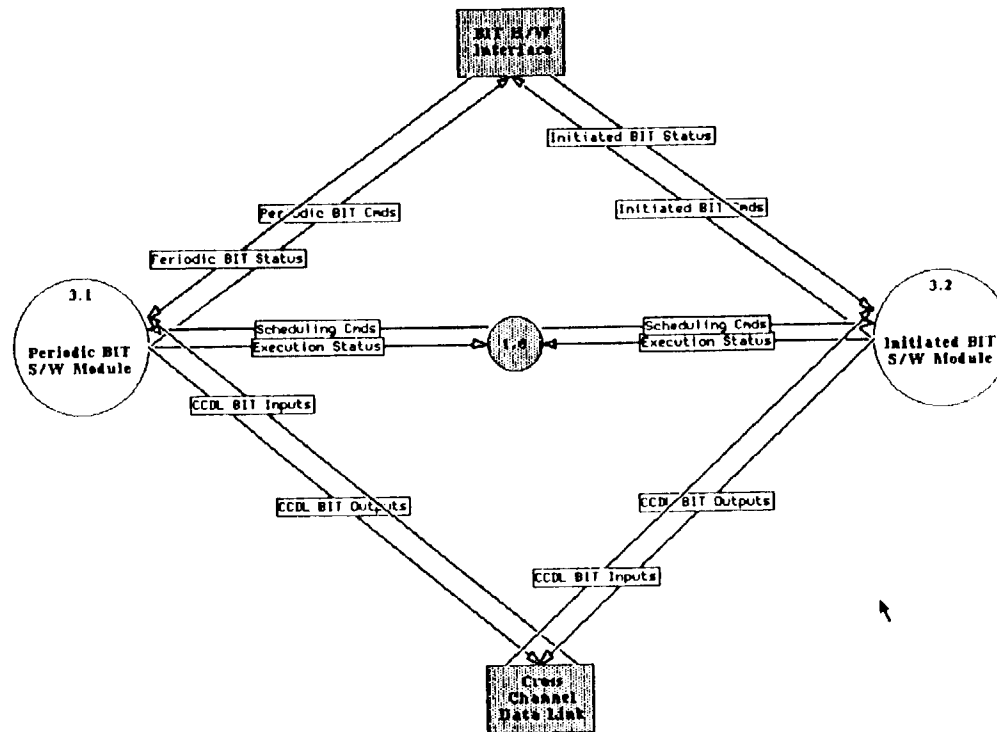


Figure 4.3-2. F/A-18A S/W Design Level 1 DFD: DFD 2.0 - Flight Control S/W Modules

F/A-18A - S/W DESIGN - PROCESS OBJECT 3.0

BIT S/W Modules



F/A-18A - S/W DESIGN - PROCESS OBJECT 3.0

BIT S/W Modules --- Mouse a picture or the background to invoke m

Figure 4.3-3. F/A-18A S/W Design Level 1 DFD: DFD 3.0 - BIT S/W Modules

# F/A-18A - S/W DESIGN - TOP LEVEL DIAGRAM

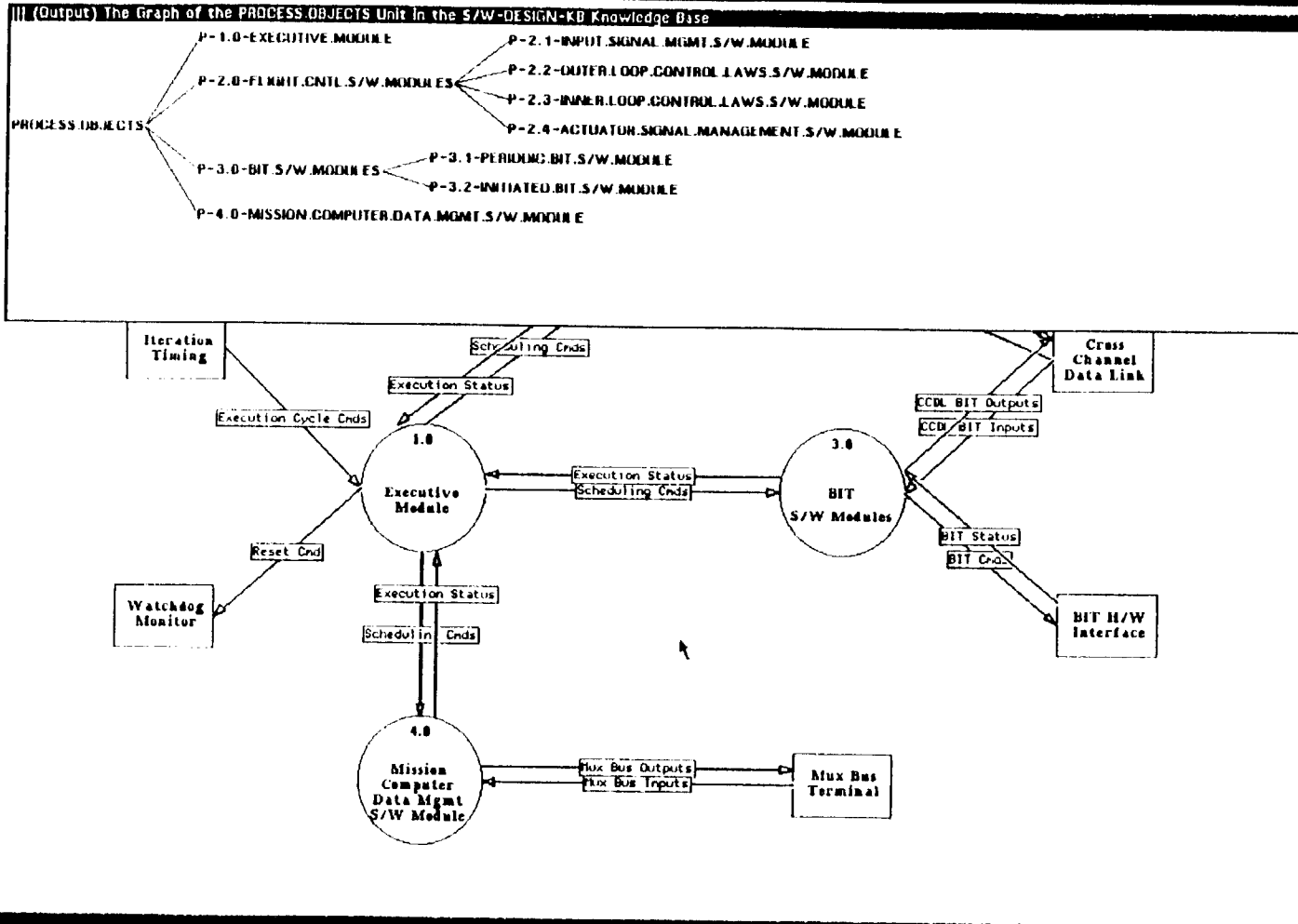


Figure 4.3-4. F/A-18A S/W Design Hierarchy of Process Objects



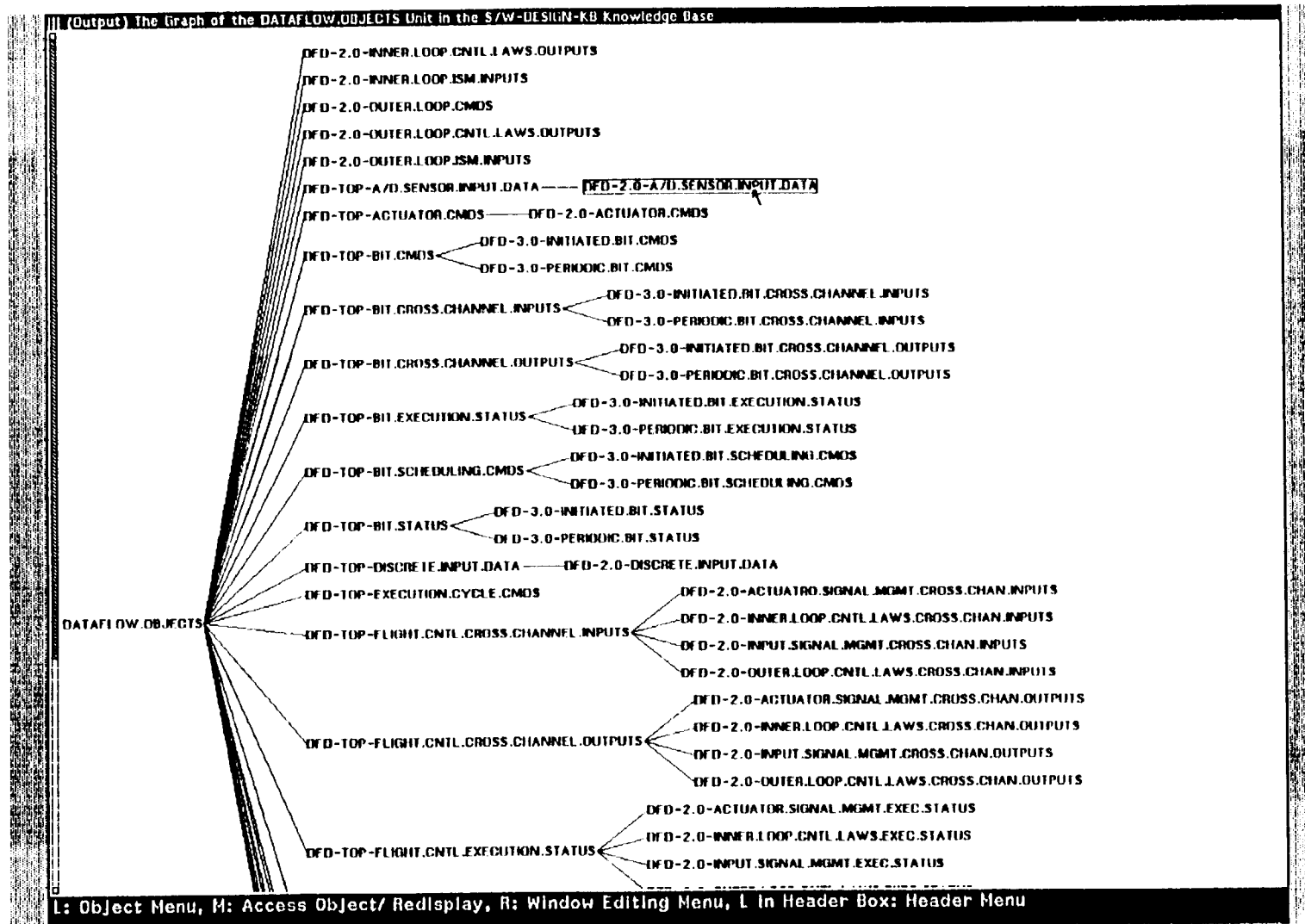


Figure 4.3-5. F/A-18A S/W Design Hierarchy of Data Flow Objects

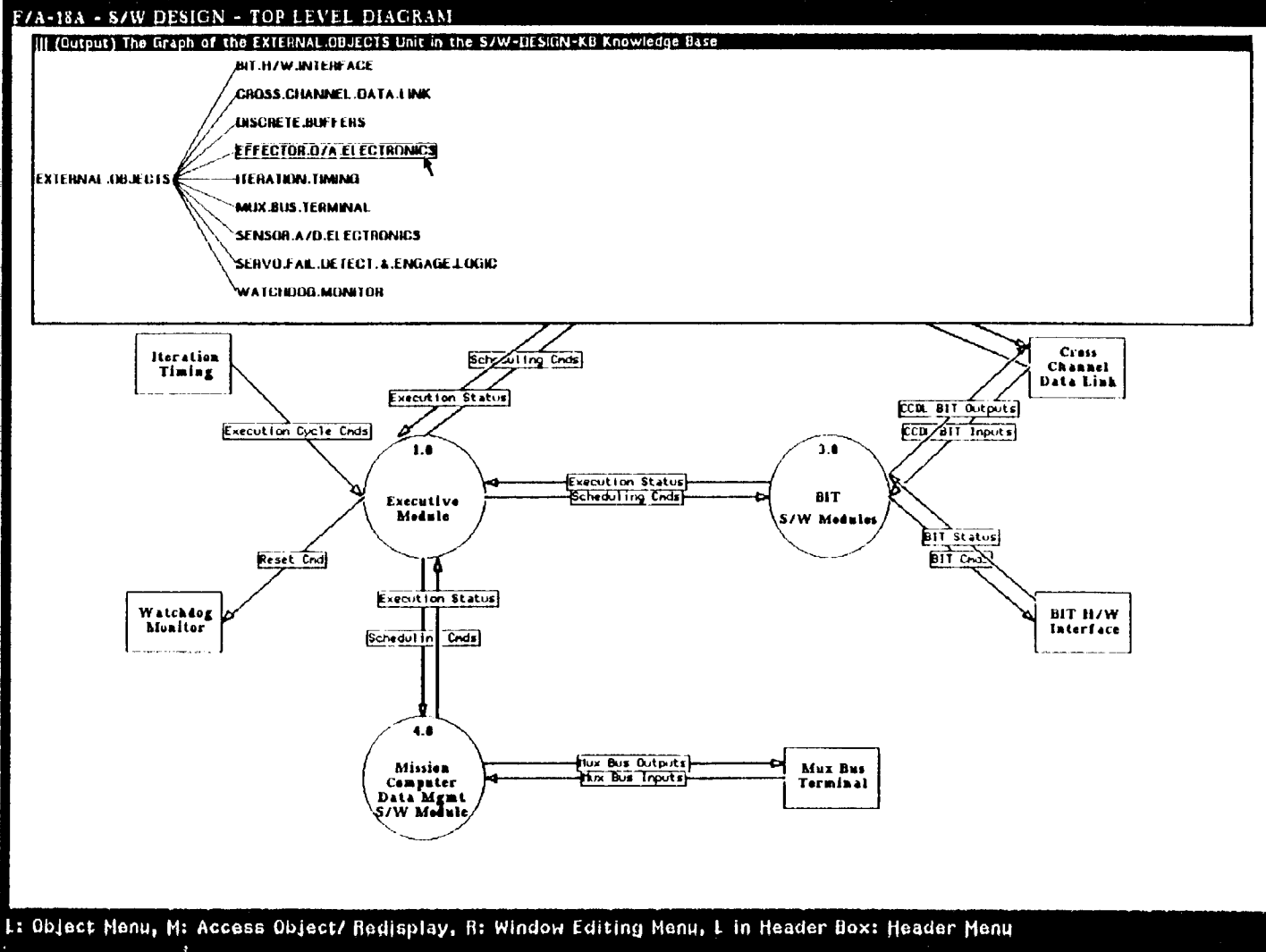


Figure 4.3-6. F/A-18A S/W Design Hierarchy of External Objects

#### **4.4 The Utilities Design Realm**

The following goals guided the selection of the FCS design which was captured in the utilities design realm.

- Demonstrate the nature of utilities design knowledge
- Demonstrate schematic capture
- Demonstrate bottom up development of the semantic network
- Demonstrate rule authoring aids
- Demonstrate behavior modeling

During the development of the KCS, the utilities design realm has been reserved for the development of bottom up design mechanisms; namely, capabilities for schematic capture, bottom up aggregation, aided rule authoring and behavior modeling. Only the top level diagram has been specified in the utilities design realm using the KCS top down decomposition mechanisms. Further FCS specification in this realm is restricted to functionality which focuses on the KCS bottom up aggregation mechanisms.

The top level utilities H/W design, which is depicted in Figure 4.4-1, identifies the major utilities H/W objects for the FCS. They include the standard F/A-18A electrical and hydraulic systems. These systems have not been decomposed in the utilities design realm, however the H/W design decomposition process and mechanisms described in Section 4.2 are applicable to these two systems.

In a conventional F/A-18A, the electrical and hydraulic systems would comprise the entire utilities design realm for this aircraft. However, the HARV has augmented the conventional electrical and hydraulic systems with an emergency system to support high angle-of-attack flight tests. This emergency system provides the capability to recover from a spin, should it occur during a flight test. The emergency system has been located in the utilities realm and its major subsystems are also indicated in the top level utilities H/W design diagram depicted in Figure 4.4-1.

It should be noted that since top down decomposition was prohibited in the utilities design realm, none of the utility top level objects could be decomposed in this realm. In fact, the

### The Semantic Network

emergency Spin Chute System has been decomposed in the H/W design realm. This decomposition is discussed in Section 5.2. Movement of this design information to the utilities design realm is considered to be a Phase 2 effort.

The semantic network hierarchy for the utilities design realm H/W objects defined by the top down decomposition are depicted in Figures 4.4-2. This hierarchy forms a parent/child relationship with inheritance features.

The KCS schematic capture development used nine individual hardcopy emergency system schematics as the bottom level definition of the implementation of the emergency system. All of these schematics were scanned and entered into the KCS as bitmap files. These schematics form a library. Four of these schematics have been aggregated to form the bottom level definition of the emergency hydraulic system (EHS) and are depicted in Figure 4.4-3. Figures 4.4-4 through 4.4-7 depict these same four schematics after the zooming feature has been used to make the components more legible.

Figures 4.4-8 through 4.4-11 depict the second of the four EHS schematics after the zooming feature has been used to further increase the component legibility. This zooming feature and the scroll bars give the user access to the entire four schematic description of the EHS. In Figure 4.4-8, the scroll bars indicate that this display comprises less than 10% of the EHS schematics. Note that the emergency system schematics are comprised of elements associated with other emergency subsystems besides the emergency hydraulic system. The hotspots indicate elements associated with the emergency hydraulic system which have been captured as H/W and S/F objects. Only those elements associated with the emergency hydraulic system are captured here.

The bottom up aggregation mechanisms have been used to define the EHS in the behavior model described in Section 5.3. Figures 5-28 through 5-29 depict the aggregation of these schematic representations into the H/W and S/F objects at a higher level of abstraction. The level 2 emergency hydraulic system objects, which were captured, have been used as the basis for a rule-based failure model. Section 5.3 also describes the use of rule authoring aids which were used to develop the emergency hydraulic system behavioral model.

The development of the top down decomposition and bottom up aggregation mechanisms assumes that the objects defined by these two design approaches will be integrated so as to

### The Semantic Network

form a common hierarchy of objects. The restricted available resources prevented the development of the necessary mechanisms to integrate the top down hierarchy with the bottom up hierarchy. This integration is one of the proposed efforts for the phase 2 activities.

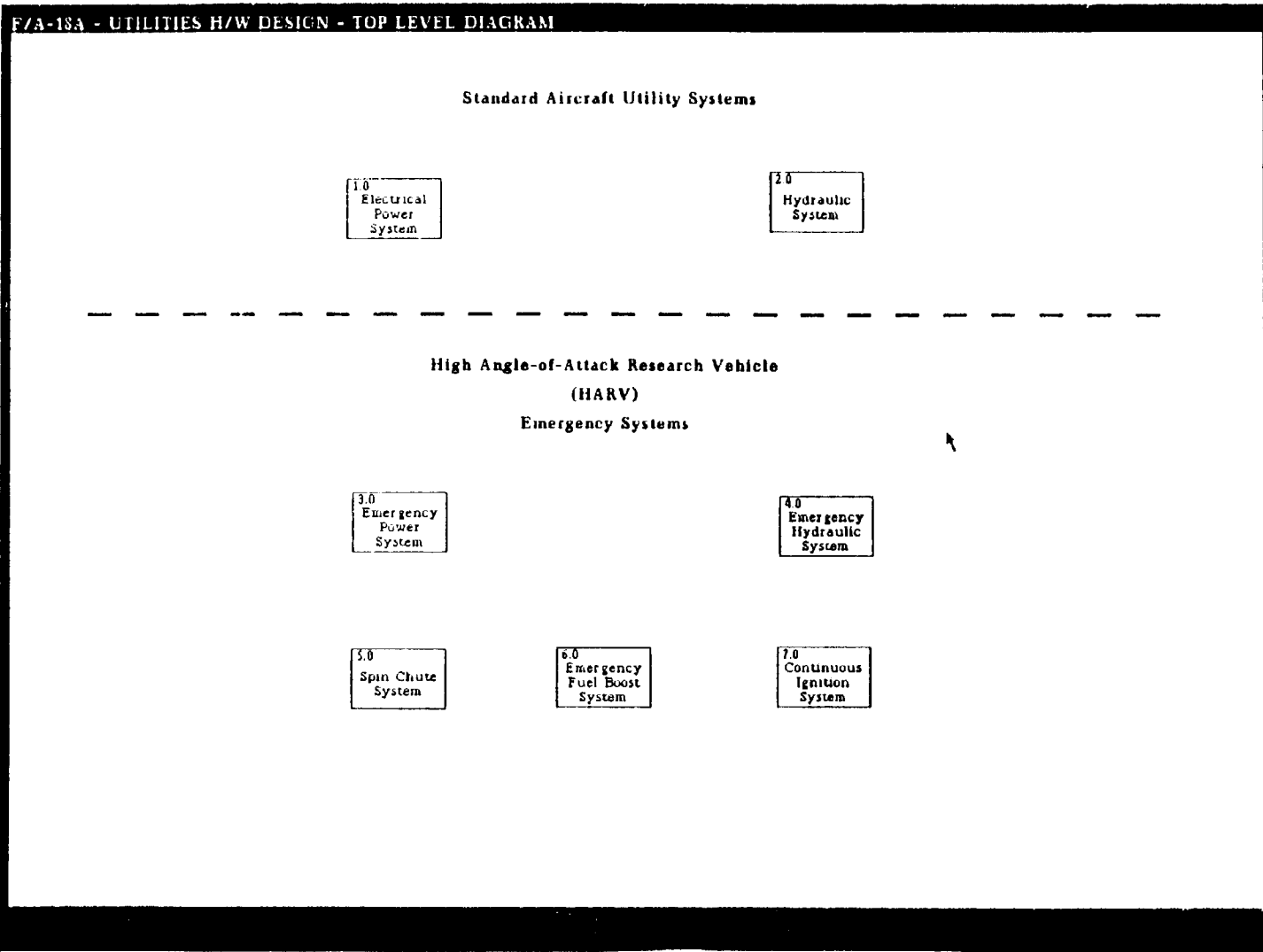


Figure 4.4-1. F/A-18A Utilities Design Level 0 H/W Diagram - The Top Level H/W Diagram

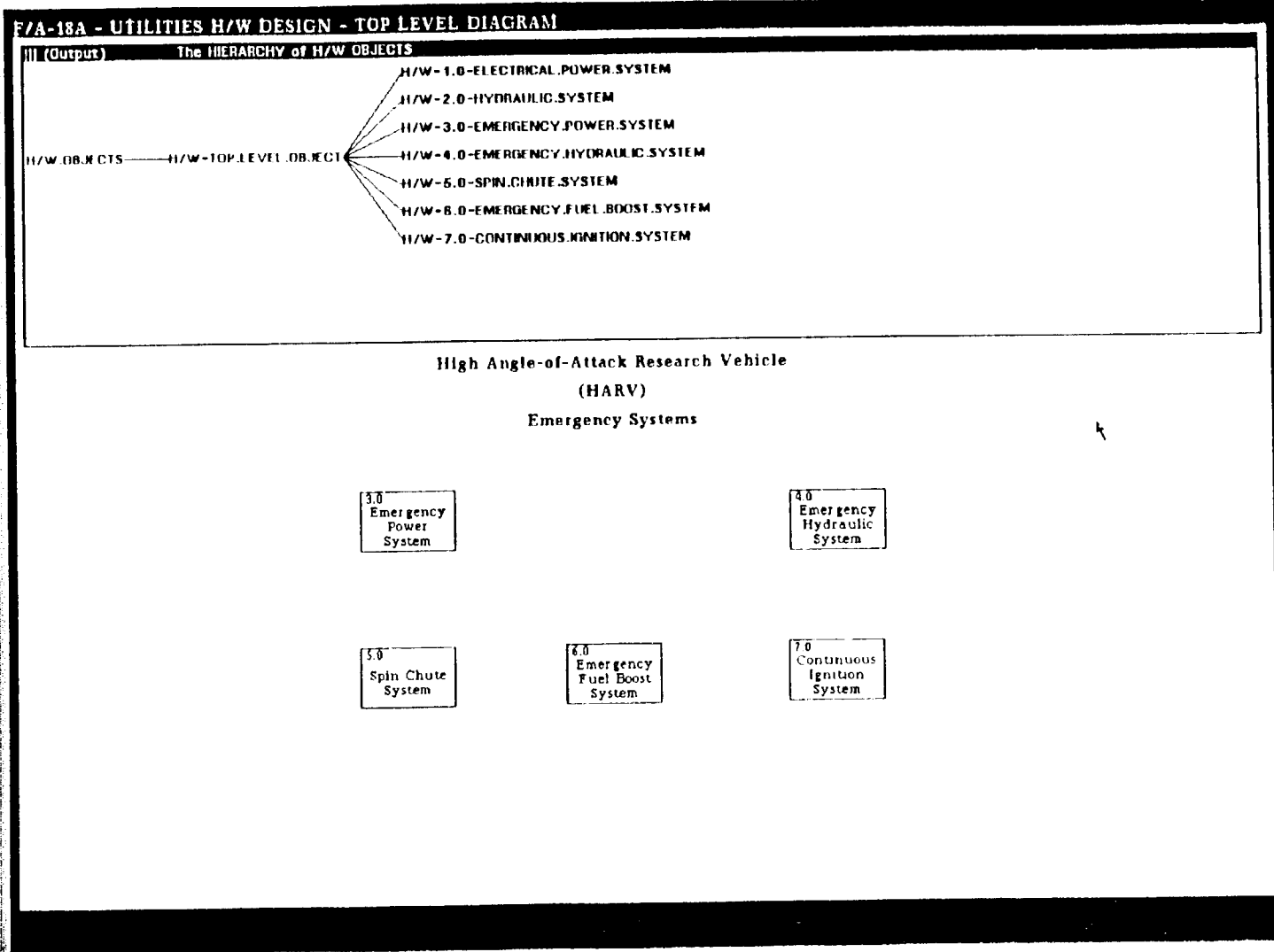


Figure 4.4-2. F/A-18A Utilities Design Top Down Hierarchy of H/W Objects

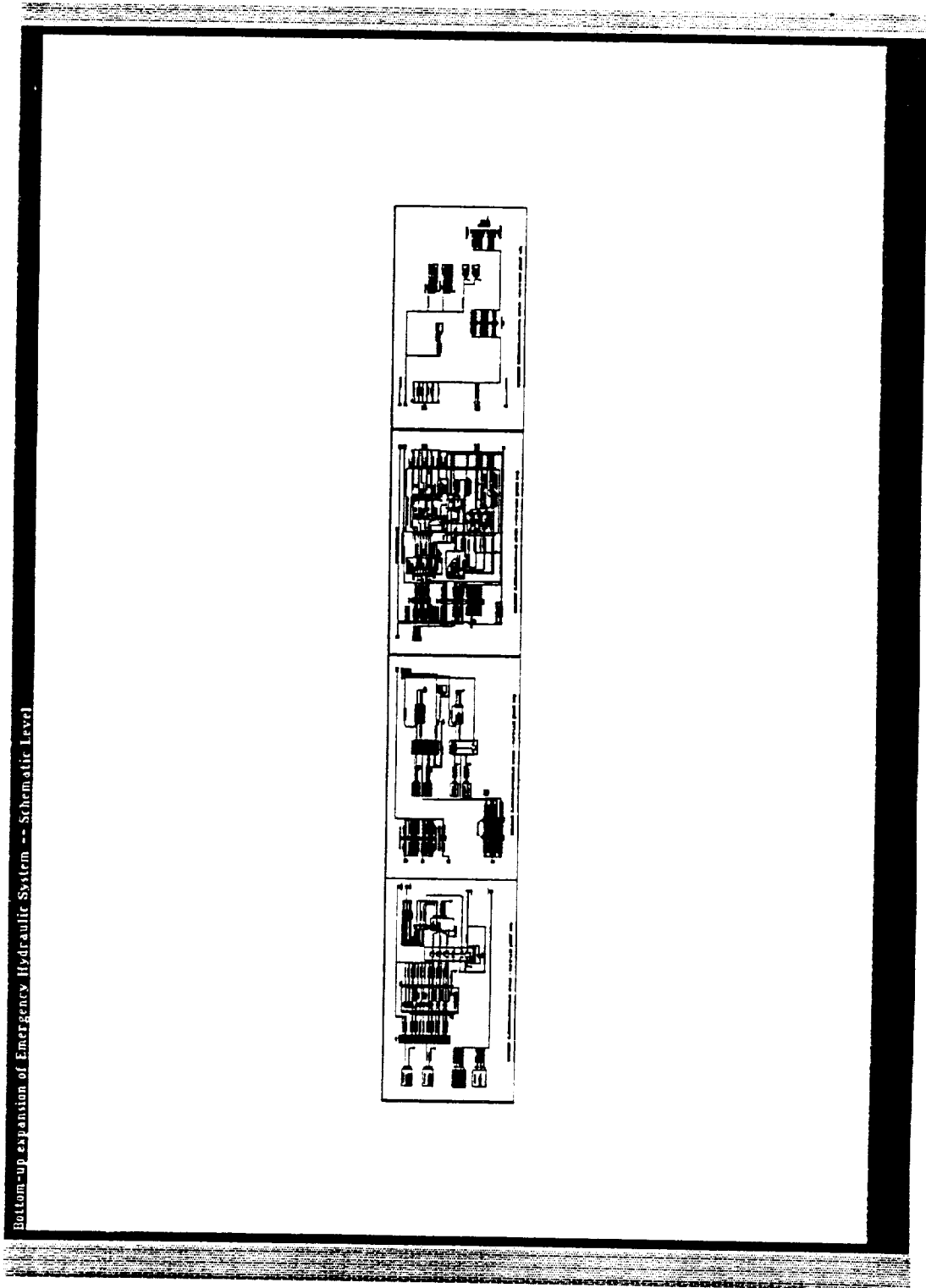


Figure 4.4-3. HARV Emergency Hydraulic System Schematic



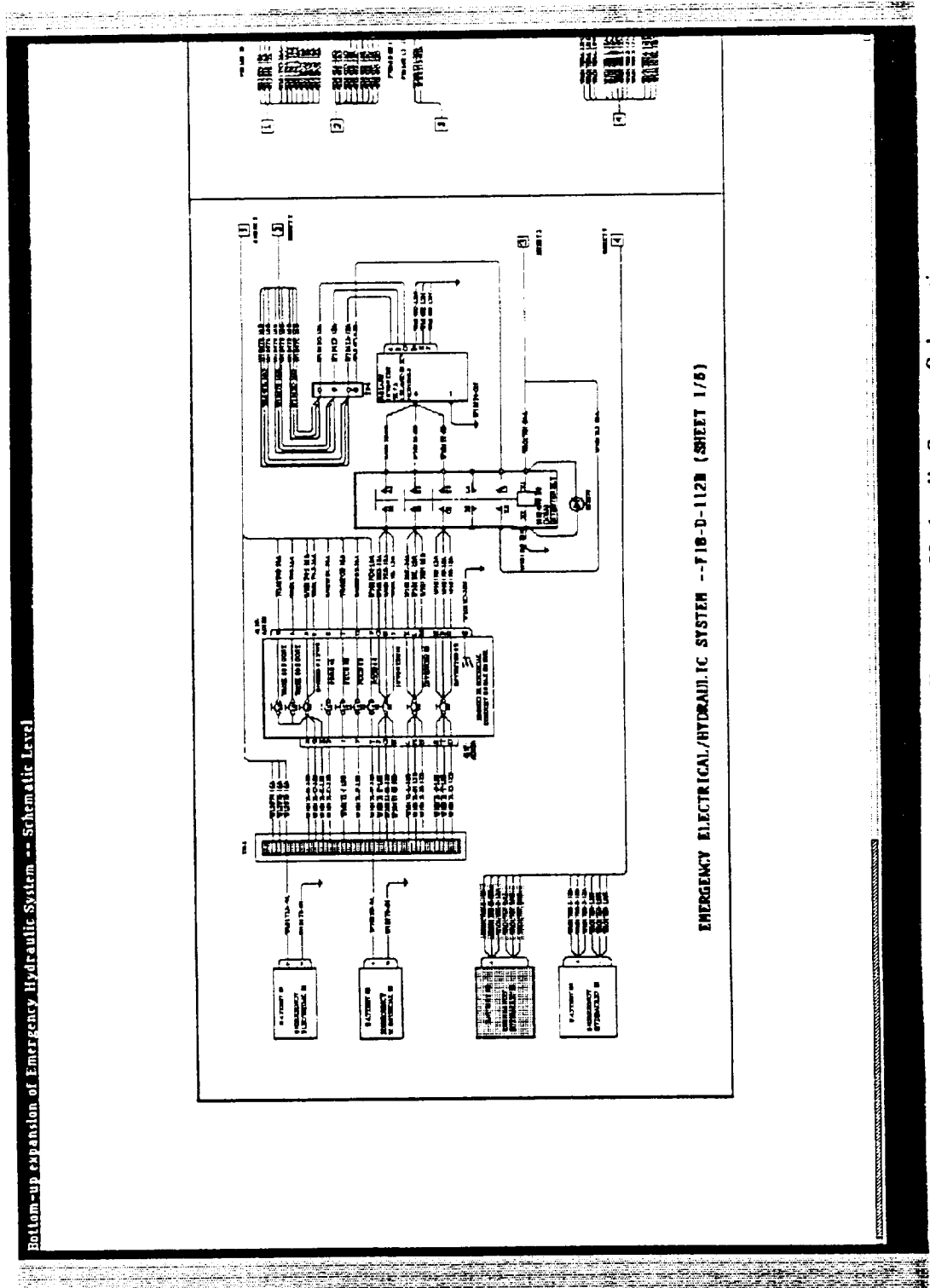
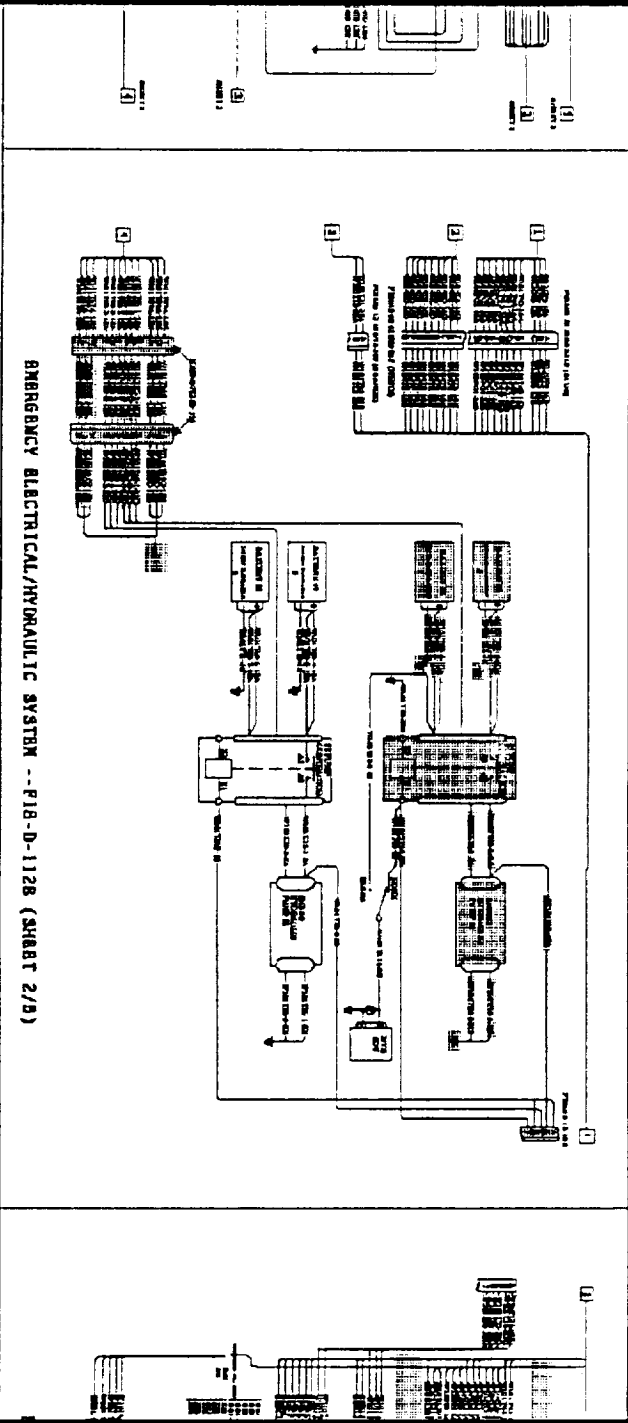


Figure 4.4-4. Sheet # 1 of the Emergency Hydraulic System Schematic

Bottom-up expansion of Emergency Hydraulic System -- Schematic Level



EMERGENCY ELECTRICAL/HYDRAULIC SYSTEM -- F18-D-112B (SHEET 2/5)

Figure 4.4-5. Sheet # 2 of the Emergency Hydraulic System Schematic

Bottom-up expansion of Emergency Hydraulic System -- Schematic Level

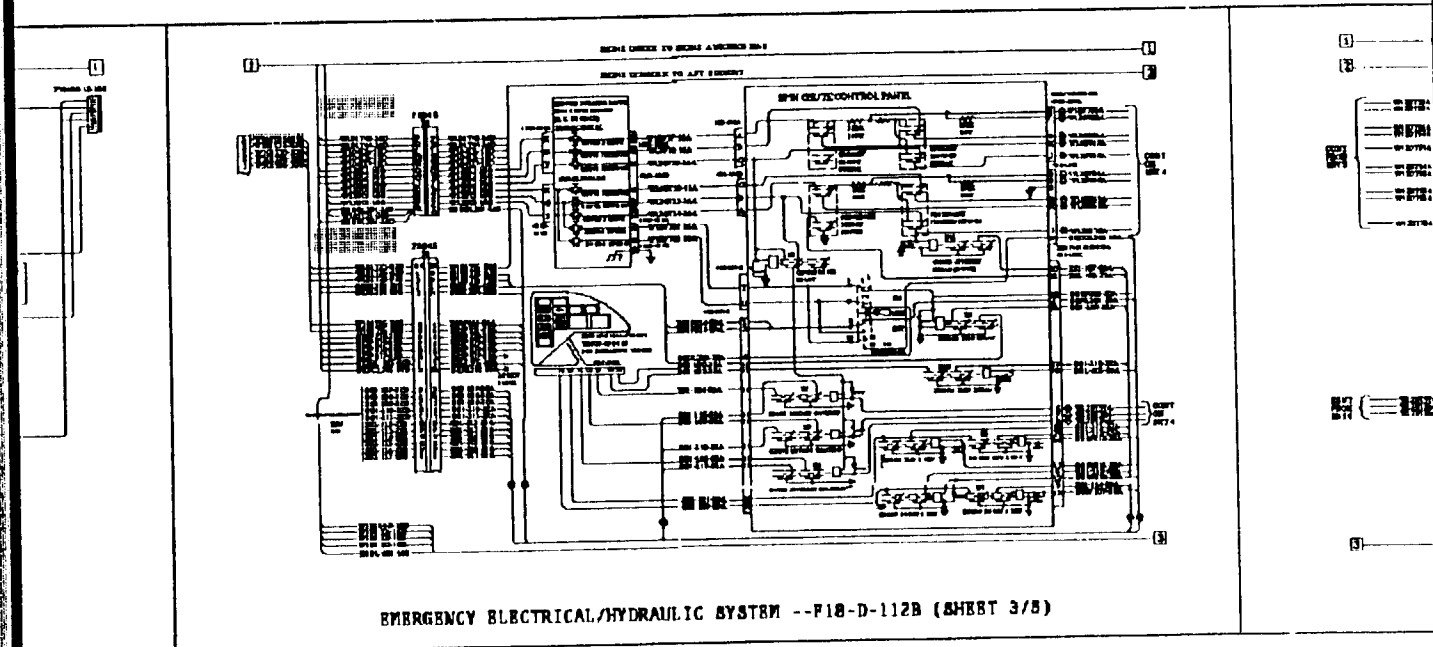


Figure 4.4-6. Sheet # 3 of the Emergency Hydraulic System Schematic

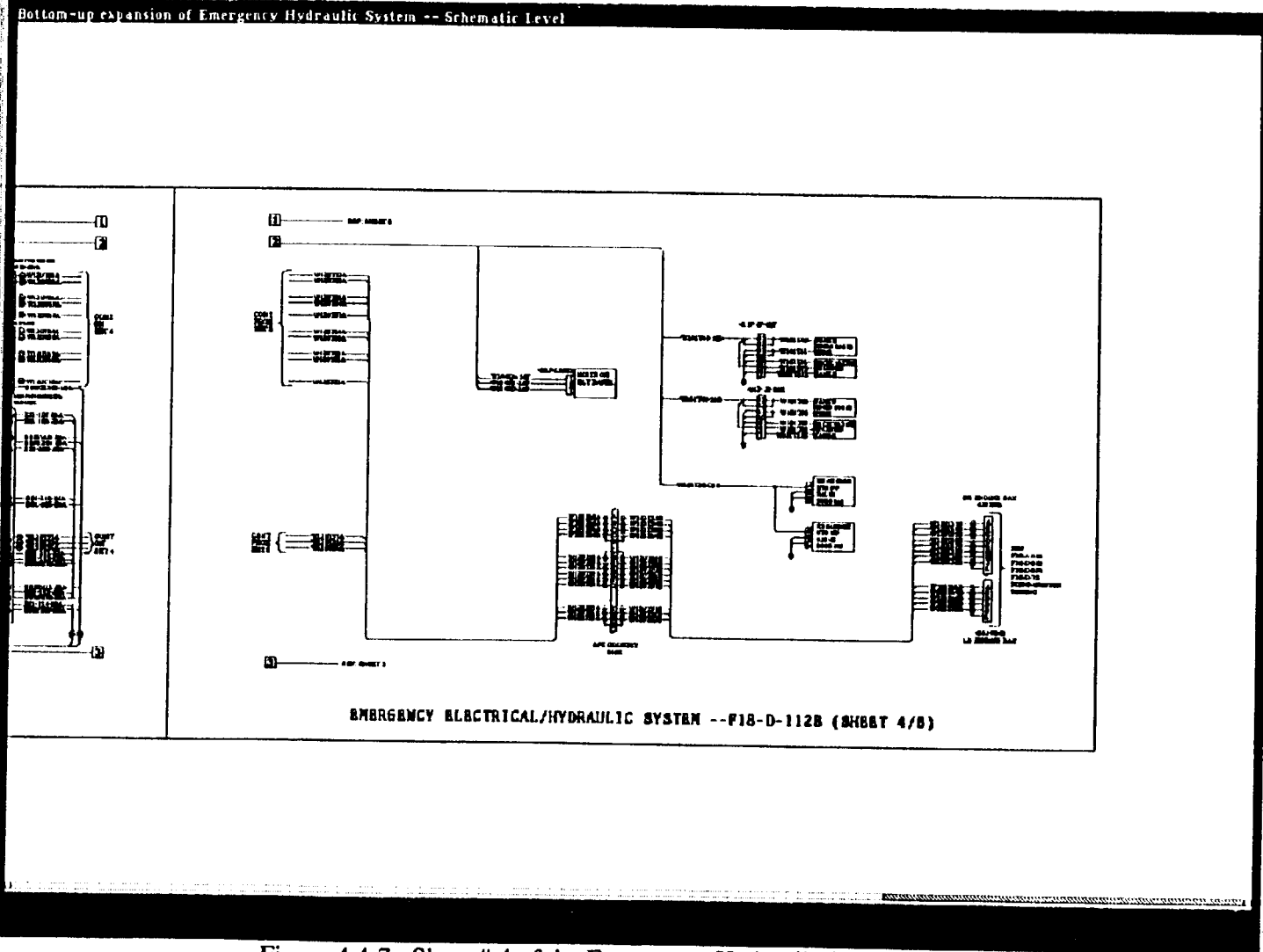


Figure 4.4-7. Sheet # 4 of the Emergency Hydraulic System Schematic

Bottom-up expansion of Emergency Hydraulic System -- Schematic Level

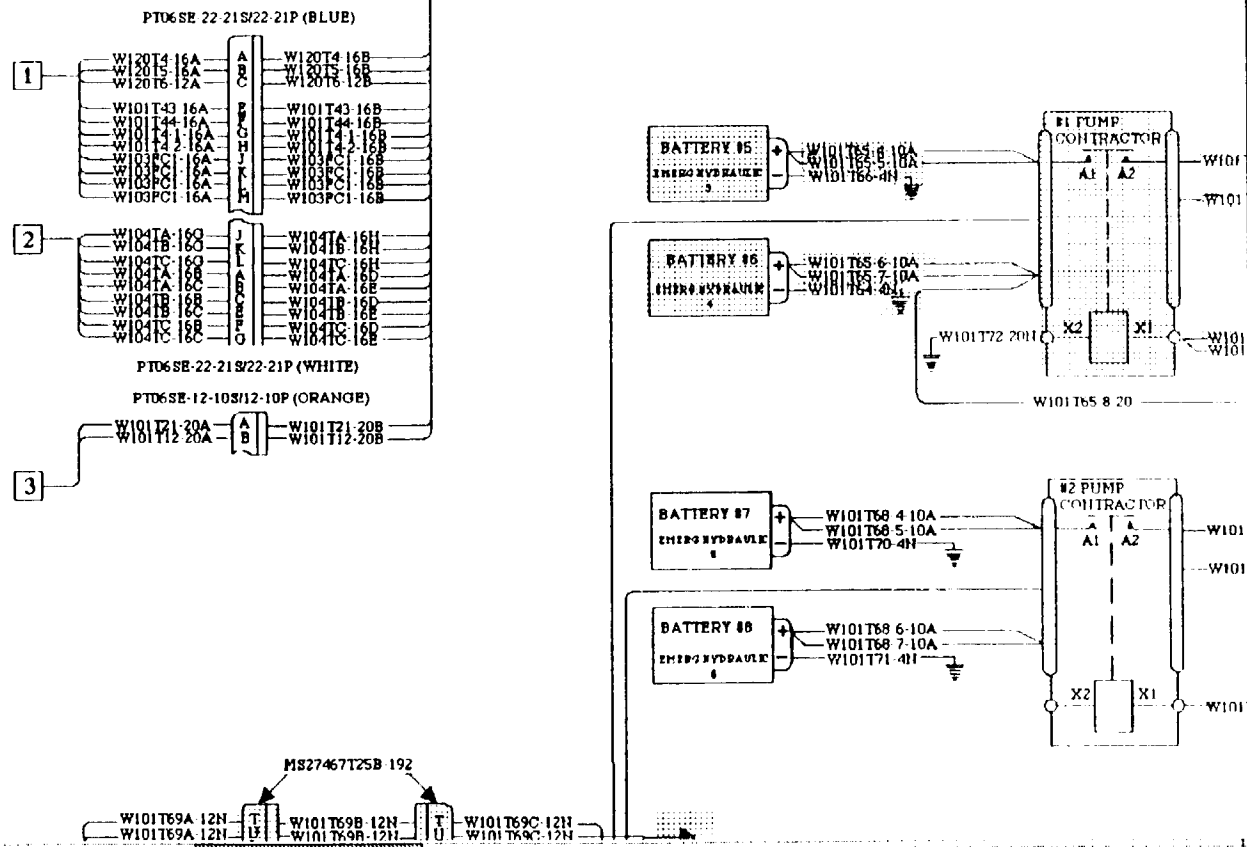


Figure 4.4-8. Upper Left Quadrant of Sheet # 2 of the Emergency Hydraulic System Schematic

4-46

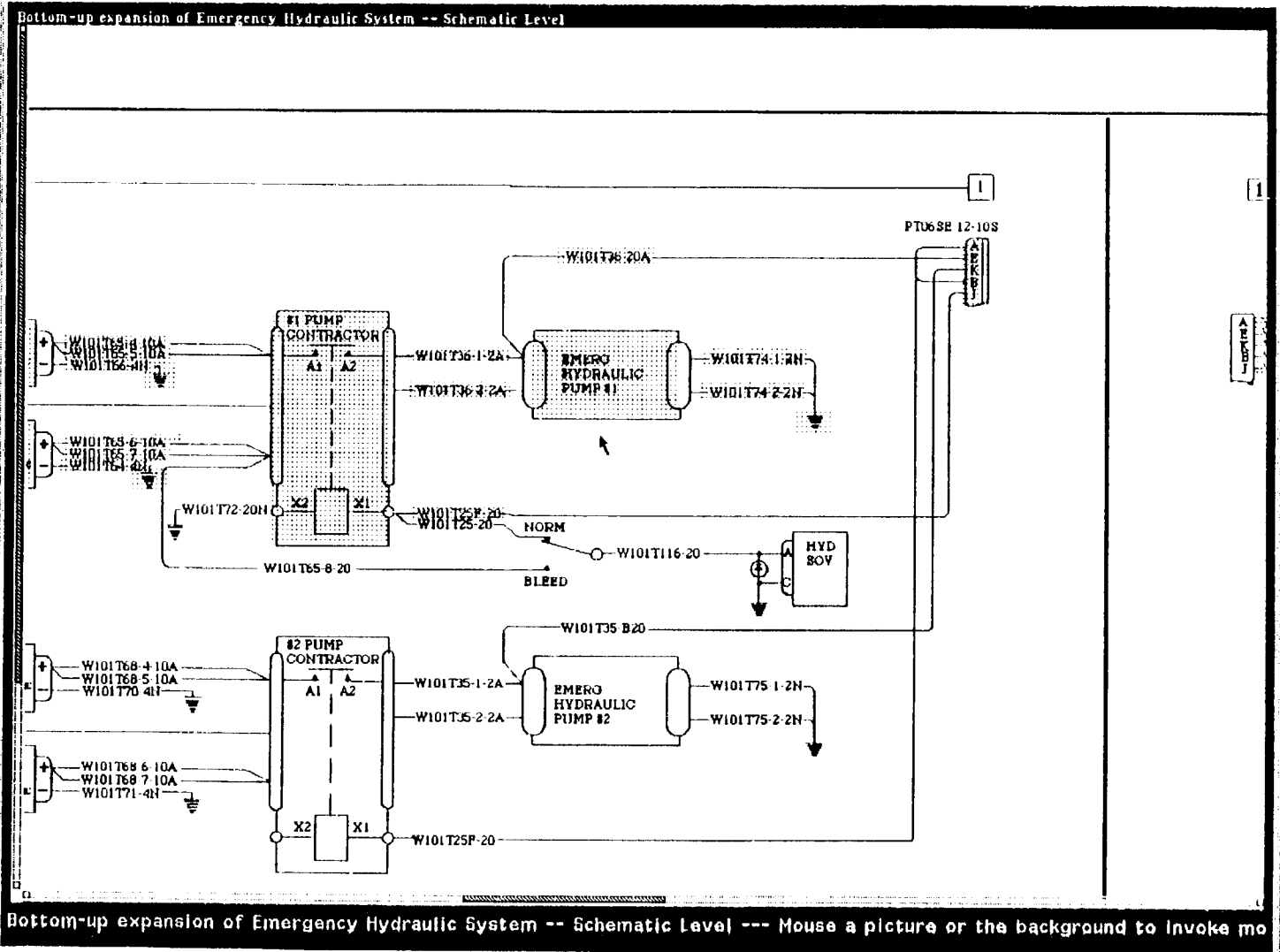


Figure 4.4-9. Upper Right Quadrant of Sheet # 2 of the Emergency Hydraulic System Schematic



Figure 4.4-10. Lower Left Quadrant of Sheet # 2 of the Emergency Hydraulic System Schematic

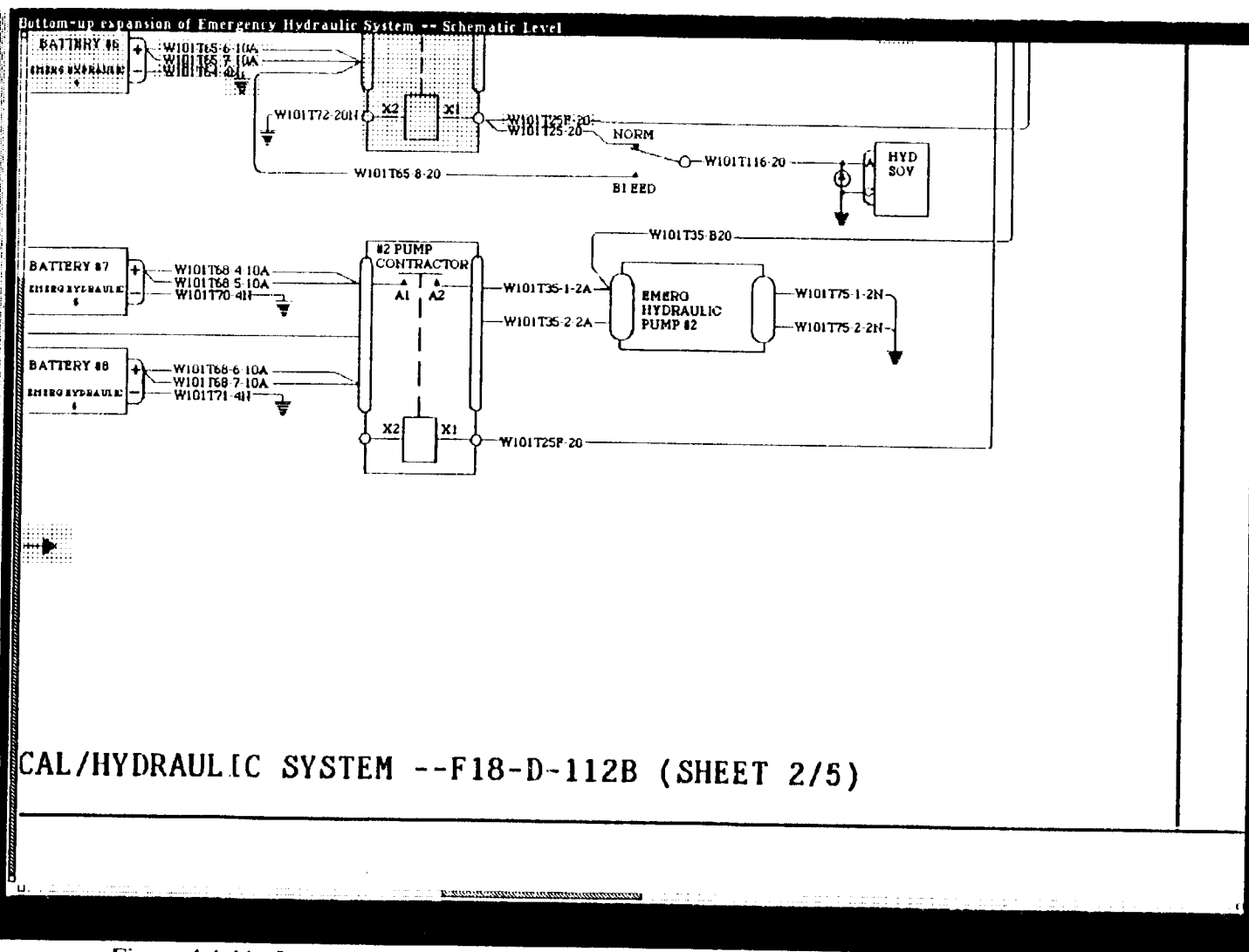


Figure 4.4-11. Lower Right Quadrant of Sheet # 2 of the Emergency Hydraulic System Schematic



## Rule-Based Models

### **5.0 Rule-Based Models**

A few models have been developed which explore the utility of the inference engine. These models focus on the behavior of a system. The forward chaining paradigm is used to dynamically indicate *"How does this system work?"* and *"What is the effect of this component failure?"*. In addition, the backward chaining paradigm is used to explain and diagnose the performance of a system. In particular it has been used to indicate *"Why won't this system work?"* and *"How redundant is this system?"*

The nose wheel steering behavioral model described here is an exploratory prototype. It permits the user to issue cockpit mode commands to a model which indicates the FCS response and changes in the aircraft state. The model is based upon a rule set and a forward chaining paradigm. A dynamic display of the rules and their execution is available to dynamically document the system operation.

A spin chute deployment rule-based model has been developed which merges the objects defined with the H/W diagram decomposition authoring mechanisms with the inference engine capability. The model shows how it is possible to build Failure Modes and Effects Analyses (FMEAs) and Fault Tree Analyses (FTAs).

A rule-based model of the emergency hydraulic system has been developed that utilizes the schematic capture mechanism. This model shows how it is possible to capture knowledge with the KCS bottom-up design capture mechanisms. In addition, it demonstrates the rule development aids which help the user develop the rules necessary for FMEAs and FTAs.

#### **5.1 Nose Wheel Steering System**

The Nose Wheel Steering (NWS) is a secondary control system which is only operable on the ground. It provides nose wheel angular deflection proportional to pedal force when engaged. There are three modes of operation: Off, Low Gain and High Gain. The desired mode is selected by the pilot with switches which are located on the control stick grip (see Figure 5-1). The NWS switch is used for NWS engagement and mode control. The autopilot disengage switch is used for NWS disengagement on the ground.

## Rule-Based Models

The low gain steering is implemented with a third order polynomial which provides a + and - 16° nose wheel deflection for maximum pedal force. The high gain steering is implemented with a fifth order polynomial which provides a + and - 75° nose wheel deflection for maximum pedal force.

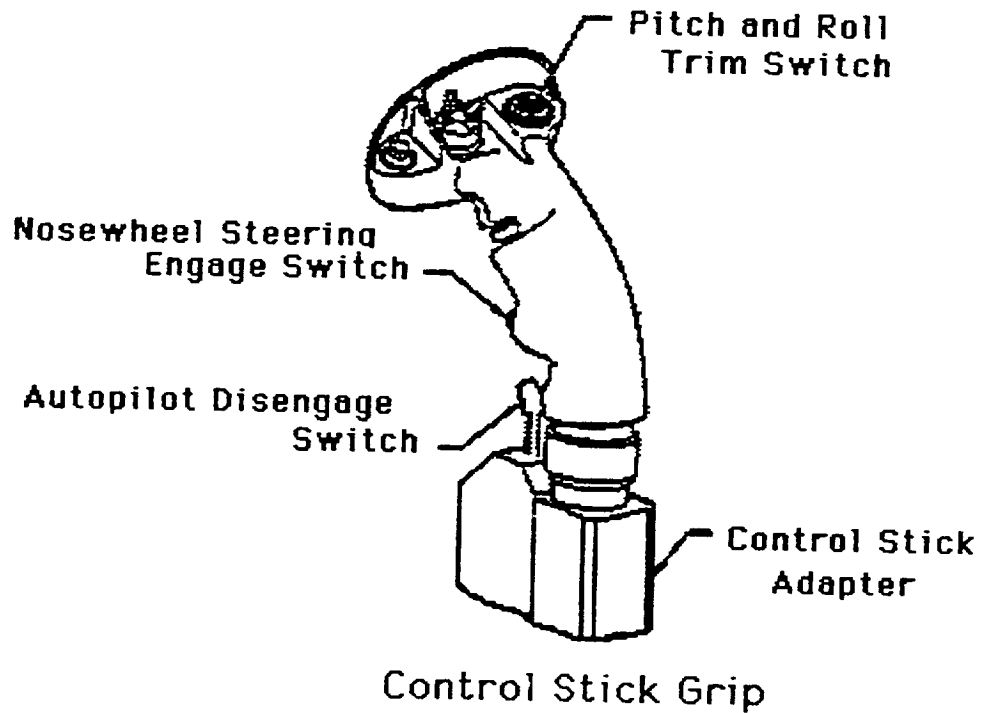


Figure 5-1. Control Stick Grip

### 5.1.1 Mode Logic Description

The control for the NWS depends on the operational mode of the aircraft. The following indicates the NWS mode logic:

- All NWS Modes
  - Nose wheel steering will not automatically engage on the ground when power is turned on.
  - Momentarily depressing the autopilot disengage switch will disengage nose wheel steering.

## Rule-Based Models

- Taxi, Take Off (T/O) and Landing Operation
  - Nose wheel steering automatically engages in the low gain mode at touchdown, but can be disengaged with momentary autopilot disengage switch operation.
  - The low gain mode engages with momentary NWS switch depression.
  - Holding the NWS switch depressed will engage the high gain mode. Reversion to the low gain mode will occur when the NWS switch is released.
- Launch and Wing Folded Operation
  - Nose wheel steering disengages when the launch bar is down unless the NWS switch is depressed which then engages the low gain mode.
  - If the wings are folded, a momentary depression of the NWS switch will engage the high gain mode. Reversion to the low gain mode will occur if the wings are subsequently spread.

### **5.1.2 The Rule-Based Model**

State variables for the autopilot disengage switch (A/P.SWITCH), the NWS switch (NWS.SWITCH), aircraft power (A/S.POWER) and aircraft operational mode (OPERATIONAL.MODE) have been defined for the model. The NWS mode logic has been modeled using these state variable definitions and a set of rules. The rules are given in Figure 5-2.

### **5.1.3 The Dynamic Behavioral Display**

The NWS System behavioral model has been designed to comply with the following goals:

- Provide a basic operational description.
- Provide a dynamic description of the NWS operational modes.

The display consists of multiple windows. Figure 5-3 depicts the display with the aircraft in a ground (or ship's deck) storage configuration. Figure 5-4 depicts the display with the aircraft in a normal configuration for taxi, take off and landing.

## Rule-Based Models

```

-----> NWS.SWITCH.HELD.DEPRESSED.CMD.RULES, whose children are:
    ---> NWS.HELD-TAXI.T/O.LANDING
    ---> NWS.HELD-LAUNCH
    ---> NWS.HELD-WINGS.FOLDED
-----

RULE UNIT NAME ---> NWS.HELD-TAXI.T/O.LANDING
(IF (THE SWITCH.STATE OF NWS.SWITCH IS HELD.DEPRESSED)
    (THE OPERATIONAL.MODE OF F/A-18A IS TAXI.T/O.LANDING)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'HIGH.GAIN.ENGAGED)))

RULE UNIT NAME ---> NWS.HELD-LAUNCH
(IF (THE SWITCH.STATE OF NWS.SWITCH IS HELD.DEPRESSED)
    (THE OPERATIONAL.MODE OF F/A-18A IS LAUNCH)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'LOW.GAIN.ENGAGED)))

RULE UNIT NAME ---> NWS.HELD-WINGS.FOLDED
(IF (THE SWITCH.STATE OF NWS.SWITCH IS HELD.DEPRESSED)
    (THE OPERATIONAL.MODE OF F/A-18A IS WINGS.FOLDED)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'HIGH.GAIN.ENGAGED)))

-----> NWS.SWITCH.MOMENTARY.CMD.RULES, whose children are:
    ---> NWS.MOM-TAXI.T/O.LANDING
    ---> NWS.MOM-WINGS.FOLDED
-----

RULE UNIT NAME ---> NWS.MOM-TAXI.T/O.LANDING
(IF (THE SWITCH.STATE OF NWS.SWITCH IS MOMENTARILY.DEPRESSED)
    (THE OPERATIONAL.MODE OF F/A-18A IS TAXI.T/O.LANDING)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'LOW.GAIN.ENGAGED)))

RULE UNIT NAME ---> NWS.MOM-WINGS.FOLDED
(IF (THE SWITCH.STATE OF NWS.SWITCH IS MOMENTARILY.DEPRESSED)
    (THE SWITCH.STATE OF A/S.POWER IS ON)
    (THE SWITCH.STATE OF A/S.TOUCHDOWN IS TRUE)
    (THE SWITCH.STATE OF A/S.WING.POSITION IS FOLDED)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'HIGH.GAIN.ENGAGED)))

-----> NWS.SWITCH.RELEASE.CMD.RULES, whose children are:
    ---> NWS.REL-LAUNCH
    ---> NWS.REL-TAXI.T/O.LANDING
-----

RULE UNIT NAME ---> NWS.REL-LAUNCH
(IF (THE SWITCH.STATE OF NWS.SWITCH IS RELEASED)
    (THE OPERATIONAL.MODE OF F/A-18A IS LAUNCH)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'DISENGAGED)))

RULE UNIT NAME ---> NWS.REL-TAXI.T/O.LANDING
(IF (THE SWITCH.STATE OF NWS.SWITCH IS RELEASED)
    (THE CONTROL.MODE OF NWS.SYSTEM IS HIGH.GAIN.ENGAGED)
    (THE OPERATIONAL.MODE OF F/A-18A IS TAXI.T/O.LANDING)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'LOW.GAIN.ENGAGED)))

-----> A/P.SWITCH.RULES, whose children are:
    ---> A/P.HELD.COMMAND
    ---> A/P.MOMENTARY.COMMAND
-----

RULE UNIT NAME ---> A/P.HELD.COMMAND
(IF (THE SWITCH.STATE OF A/P.SWITCH IS HELD.DEPRESSED)
    (THE SWITCH.STATE OF NWS.SWITCH IS RELEASED)
    (THE SWITCH.STATE OF A/S.POWER IS ON)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'DISENGAGED)))

RULE UNIT NAME ---> A/P.MOMENTARY.COMMAND
(IF (THE SWITCH.STATE OF A/P.SWITCH IS MOMENTARILY.DEPRESSED)
    (THE SWITCH.STATE OF NWS.SWITCH IS RELEASED)
    (THE SWITCH.STATE OF A/S.POWER IS ON)
    THEN
    (LISP (PUT.VALUE '(NWS.SYSTEM F-18-NWS-SYSTEM) 'CONTROL.MODE 'DISENGAGED)))

```

Figure 5-2. NWS Switch and Autopilot Switch Rules

## Rule-Based Models

A dynamic interactive display of the NWS modes is provided to control and display the following information:

- The Pilot Commands (The Control Stick Switch Commands)
- The NWS System Status
- The NWS System Block Diagram
- The Relevant F/A-18A Aircraft Status

The above information is displayed in four individual windows. The display of the Control Stick Switches includes a control stick and KEE™ active images for the NWS and autopilot switches. This display window will accept switch commands in an identical fashion to those issued by the pilot via the actual aircraft control stick. The KEE™ active images, which depict the NWS switch and the autopilot switch, are mouse sensitive. It is possible to issue a momentarily depressed, held depressed or released command with these images. The display of the aircraft control stick is also mouse sensitive.

The rule-based model implements the NWS mode logic. These rules are activated by the switch commands. The NWS System response is displayed by highlighting the appropriate mode in the NWS Control Mode Status window. The NWS System response is also displayed by highlighting the control path in the NWS system block diagram window.

The display of the aircraft status is also mouse sensitive. It is possible to explore the NWS logical operation as a function of: aircraft power, touchdown status, wing status and launch bar status by mousing the appropriate active image. As these parameters are changed, the appropriate operational mode is dynamically updated and displayed in the F/A-18A Operation Mode window. The NWS related aircraft operational modes are: power off, wings folded, taxi, take off (T/O), launch, inflight and landing.

The rule-based model is also used to implement the effects of changes in the aircraft status on the NWS mode logic. These rules are activated by changes in the aircraft status variables. The aircraft state is displayed by highlighting the appropriate value of the aircraft state variables and by displaying the current mode in the Operational Mode window. The NWS System response is displayed by highlighting the the appropriate mode in the NWS Control Mode Status window. The NWS System response is also displayed by highlighting the control path in the NWS system block diagram window.

## Rule-Based Models

It is possible to trace the rule execution in a KEE™ dynamic forward chaining execution window. The rule displays are mouse sensitive. Rule text display and rule modification is possible. A rule trace and text display are depicted in Figure 5-4. The information in this figure reflects a "momentarily depressed" command to the NWS switch.

### **5.1.4 Conclusions**

The NWS System behavioral model has indicated the usefulness of integrating a dynamic model within a KCS that contains the objects and their properties for an automated system. The behavioral model complements the simple compilation of system attributes with a dynamic model which shows how the system functions in different modes.

A useful extension or application of this type of model is the implementation a similar rule-based system for the flight control laws. These control laws are significantly more complicated than those described here for the NWS system. It is projected that such a modal behavioral model would significantly enhance the ability of the user to understand the performance of these more complicated control systems. This improvement in user understanding would enhance performance of such a user in all phases of the life cycle of the flight control system.

It is proposed that multiple behavioral models be developed which use common state variables. This implementation would interconnect the multiple behavioral models and allow the user to explore the interaction of the many subsystems which comprise the total system.

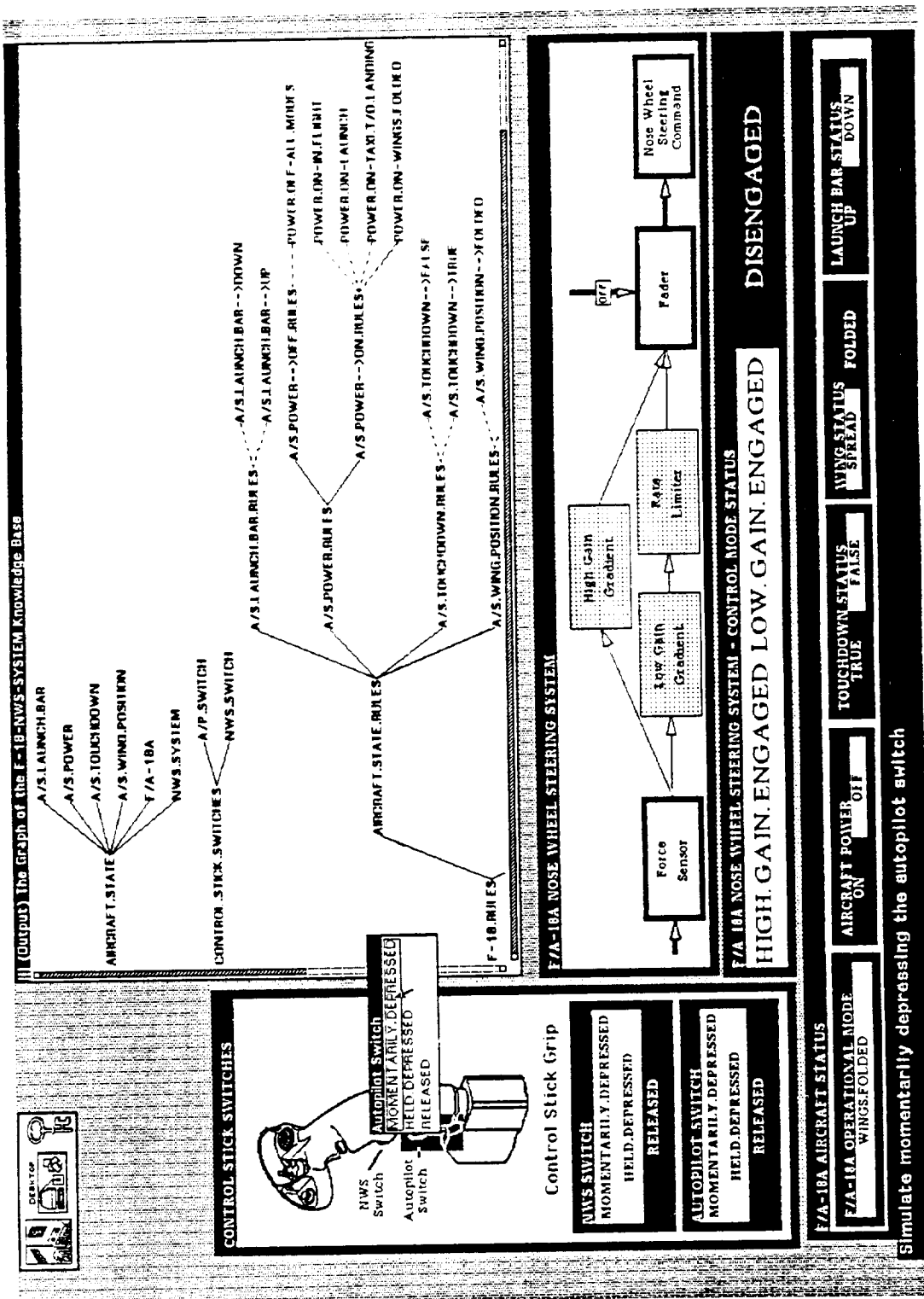


Figure 5-3. NWS Display - Wings Folded Status

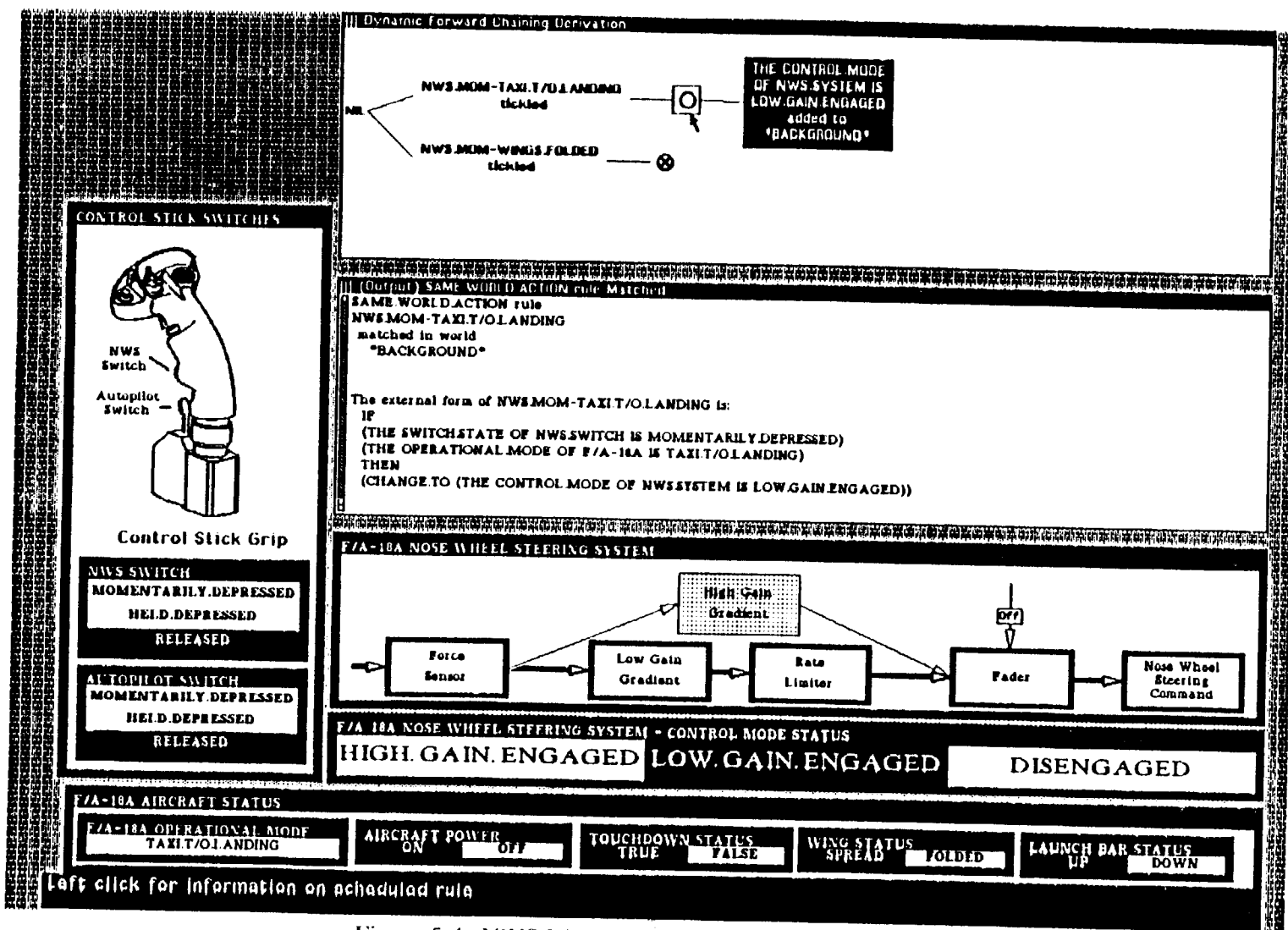


Figure 5-4. NWS Display - Taxi, T/O and Landing Status



## **5.2 Spin Chute Deployment Model**

The NASA High Angle-of-attack Research Vehicle (HARV) is a modified F/A-18A aircraft which will be flown at angles of attack above 55°. Experimental flight in this flight regime mandates the incorporation of a spin recovery system. A spin recovery parachute has been added to the F/A-18A which enables the pilot to recover from a spin. The following describes a rule-based model for the spin chute deployment system.

The spin chute deployment model uses the objects which are defined in the H/W top down decomposition found in the H/W design realm. These spin chute deployment H/W objects are incorporated into a rule-based model. When this model makes use of the forward chaining paradigm, it implements a Failure Modes and Effects Analysis (FMEA). When this model makes use of the backward chaining paradigm, it implements a Fault Tree Analysis (FTA). It can be seen that the rule-based model integrates the hierarchical frame-based decomposition with an inference engine to provide these useful behavioral models.

### **5.2.1 The H/W Decomposition**

The spin chute is deployed by the activation of pyrotechnics. Pyrotechnics release the containment cannister strap of the tail mounted spin chute cannister. Other pyrotechnics initiate a rocket which deploys the spin chute. These pyrotechnics are activated with dual redundant circuitry.

The H/W methodology is used to capture the spin chute deployment design. This spin chute deployment pyrotechnic system is decomposed into the following H/W objects.

1. H/W Object 29.1 - Primary Deployment Circuitry
2. H/W Object 29.2 - Secondary Deployment Circuitry
3. H/W Object 29.3 - Deployment Pyrotechnics

These H/W objects are then decomposed into their component parts. Figures 5-5 and 5-6 depict the hierarchical graph associated with this decomposition.

Figure 5-7 depicts the level 1 H/W diagram. Figures 5-5 through 5-10 depict the H/W diagrams which describe this decomposition at level 2. The decomposition continues to level 3 for the circuit breaker panel, control panel components and the spin chute assembly

## Rule-Based Models

components (Figures 5-11 through 5-13). It should be noted that the diagrams use a dual representation scheme. The box and line object representation scheme reflects the standard H/W methodology functionality. A bitmap representation scheme has been added to depict the circuit diagrams in a form which is more user friendly in this problem domain. This representation utilizes hotspots to link the two different representations. The highlighting in these figures illustrates this linkage. The highlighting is activated when the objects or the hotspots are moused.

The single wire, which is highlighted in the circuit diagram depicted in Figure 5-9, is implemented with a number of physical components. These components include a wire from the cockpit control panel which runs to an aft cockpit disconnect, a wire from this disconnect then runs to the rear of the aircraft to an engine bay connector and finally a wire from this connector runs to the spin chute assembly. The use of highlighting and dual representation are also illustrated in the other spin chute H/W diagrams.

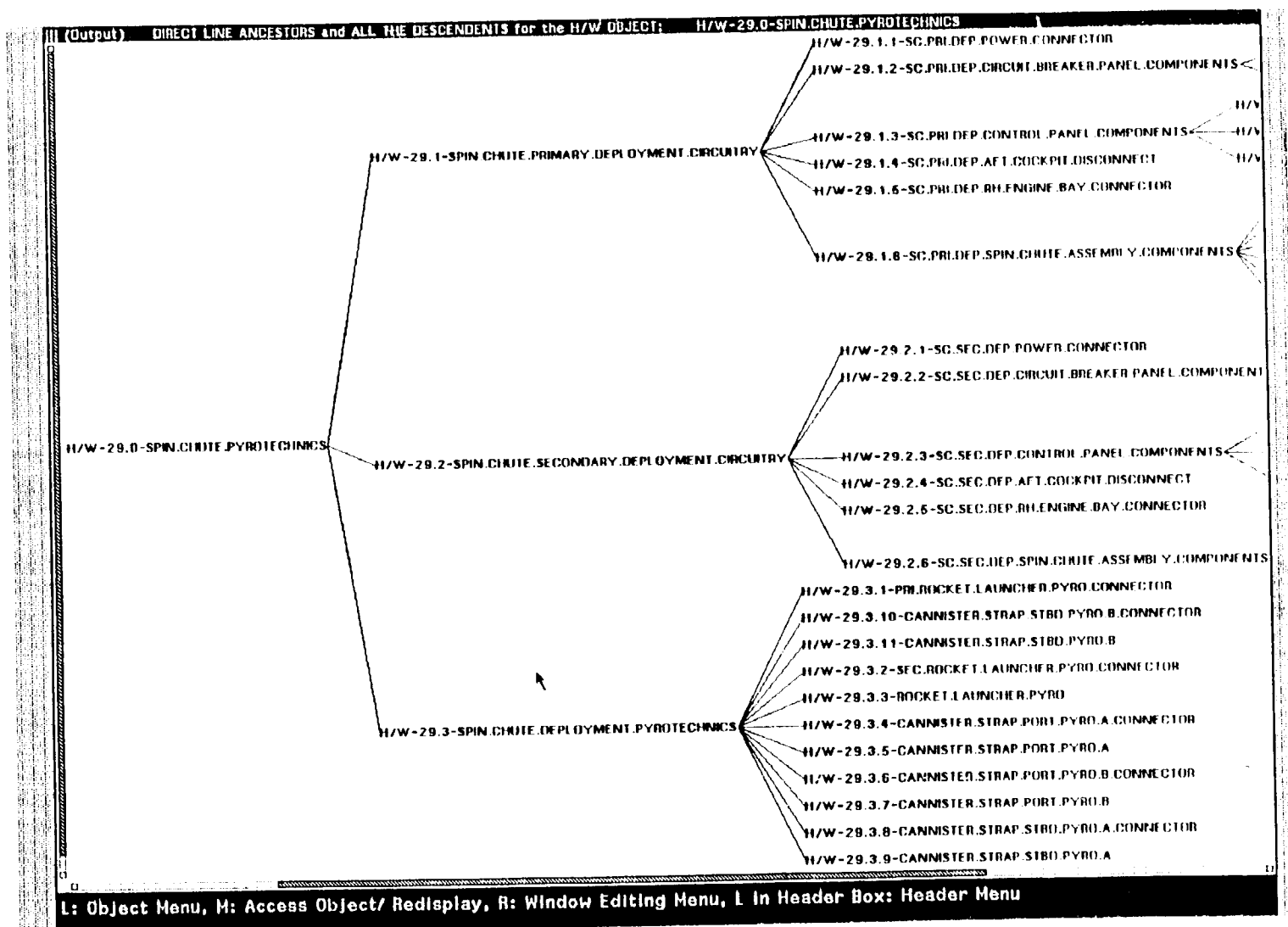


Figure 5-5. Spin Chute Deployment System Decomposition H/W Object Hierarchy (Top Levels)

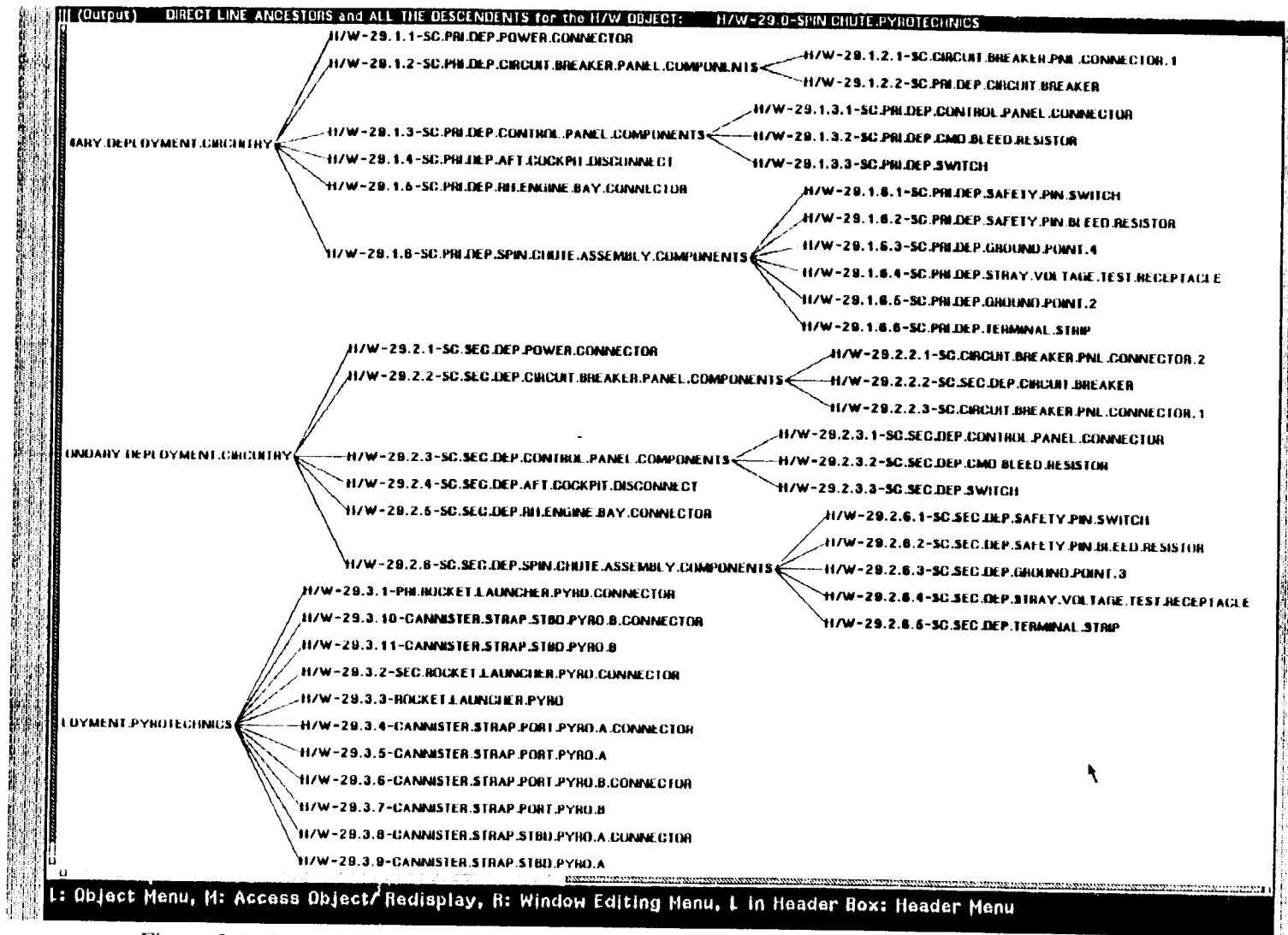


Figure 5-6. Spin Chute Deployment System Decomposition H/W Object Hierarchy (Lower Levels)

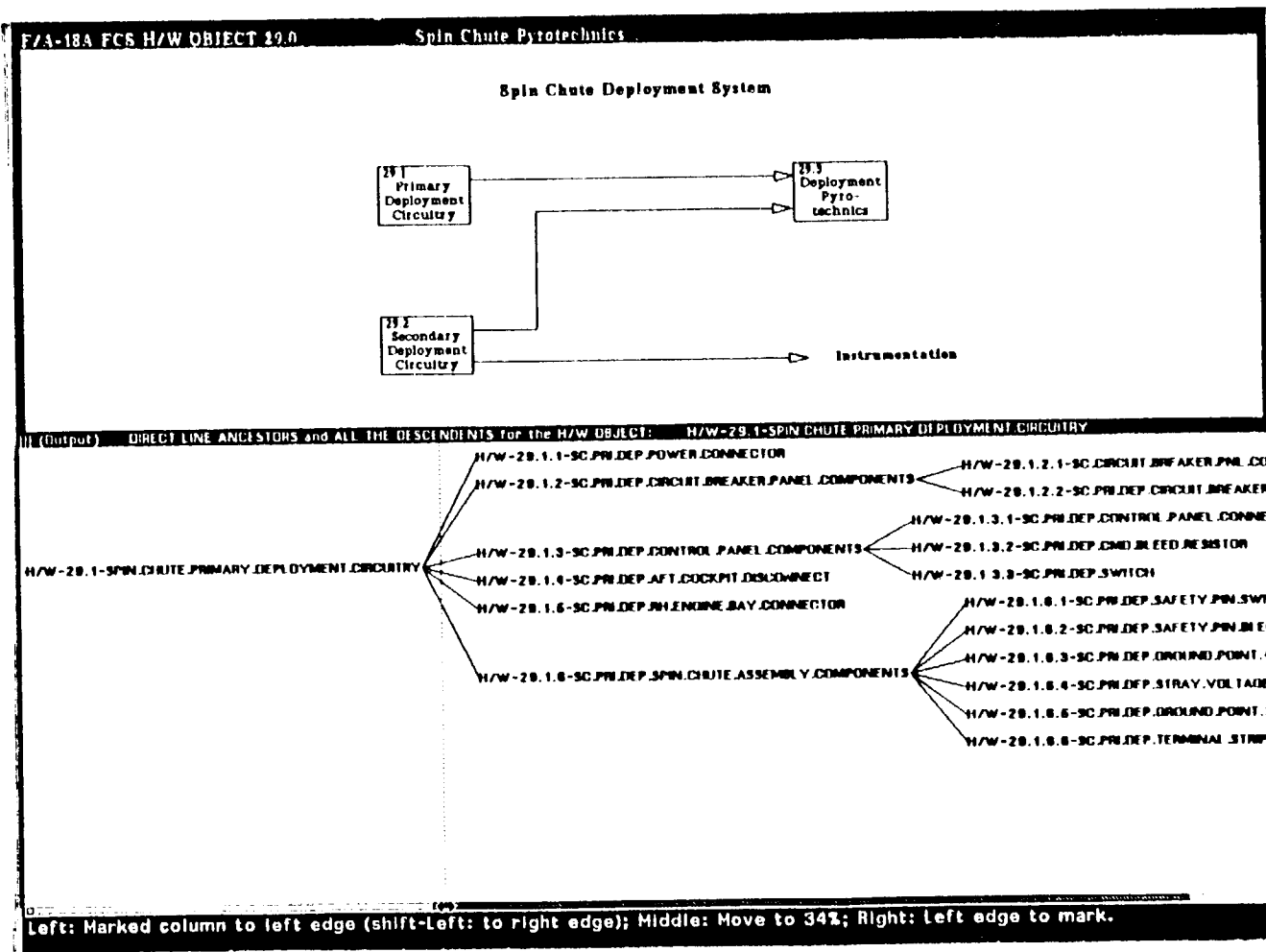


Figure 5-7. Spin Chute Deployment H/W Diagram (Level 1) and the Primary Circuit Decomposition Hierarchy.

Figure 5-8. Primary Deployment Circuit H/W Diagram (Level 2)

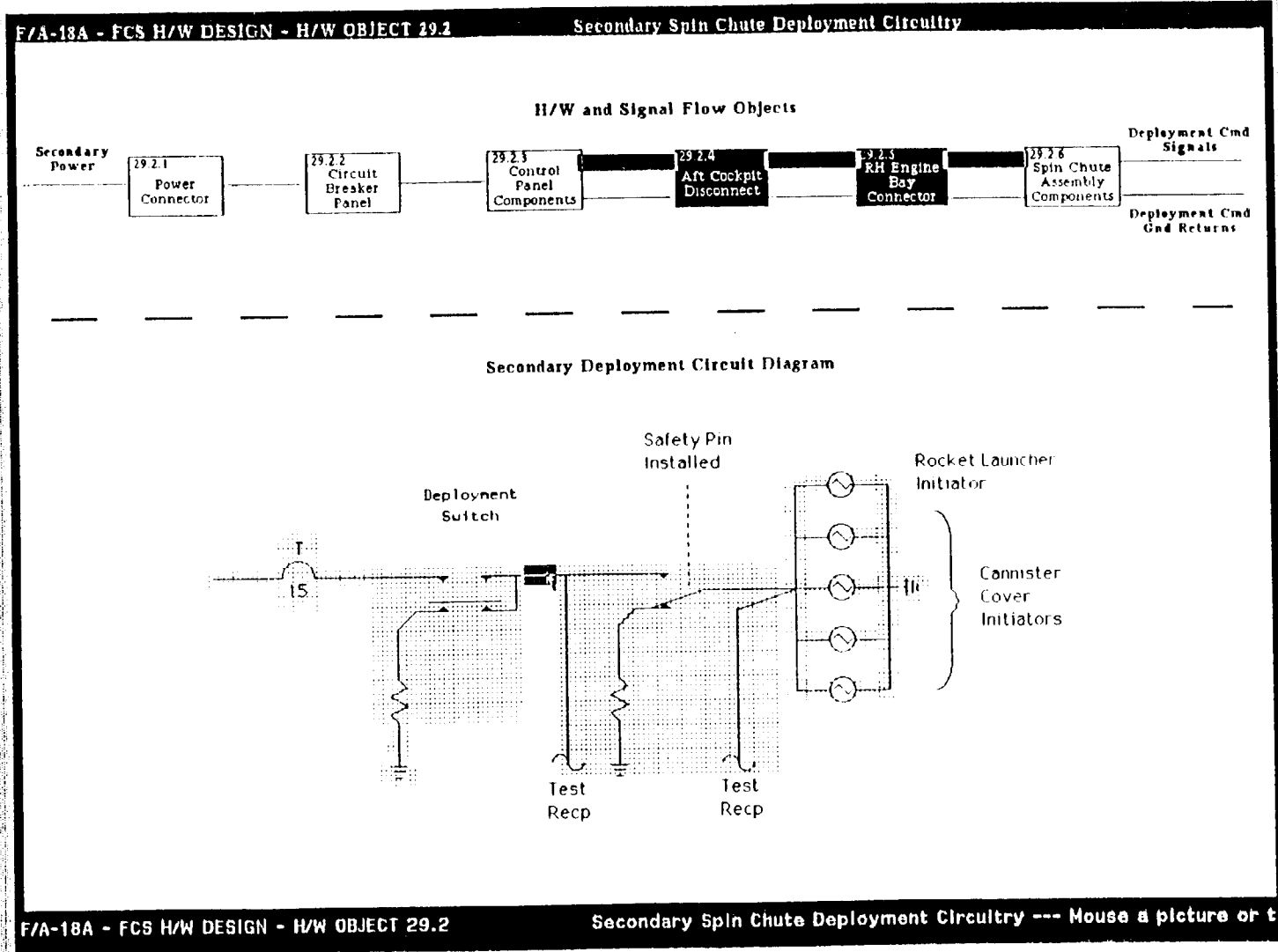


Figure 5-9. Secondary Deployment Circuit H/W Diagram (Level 2)

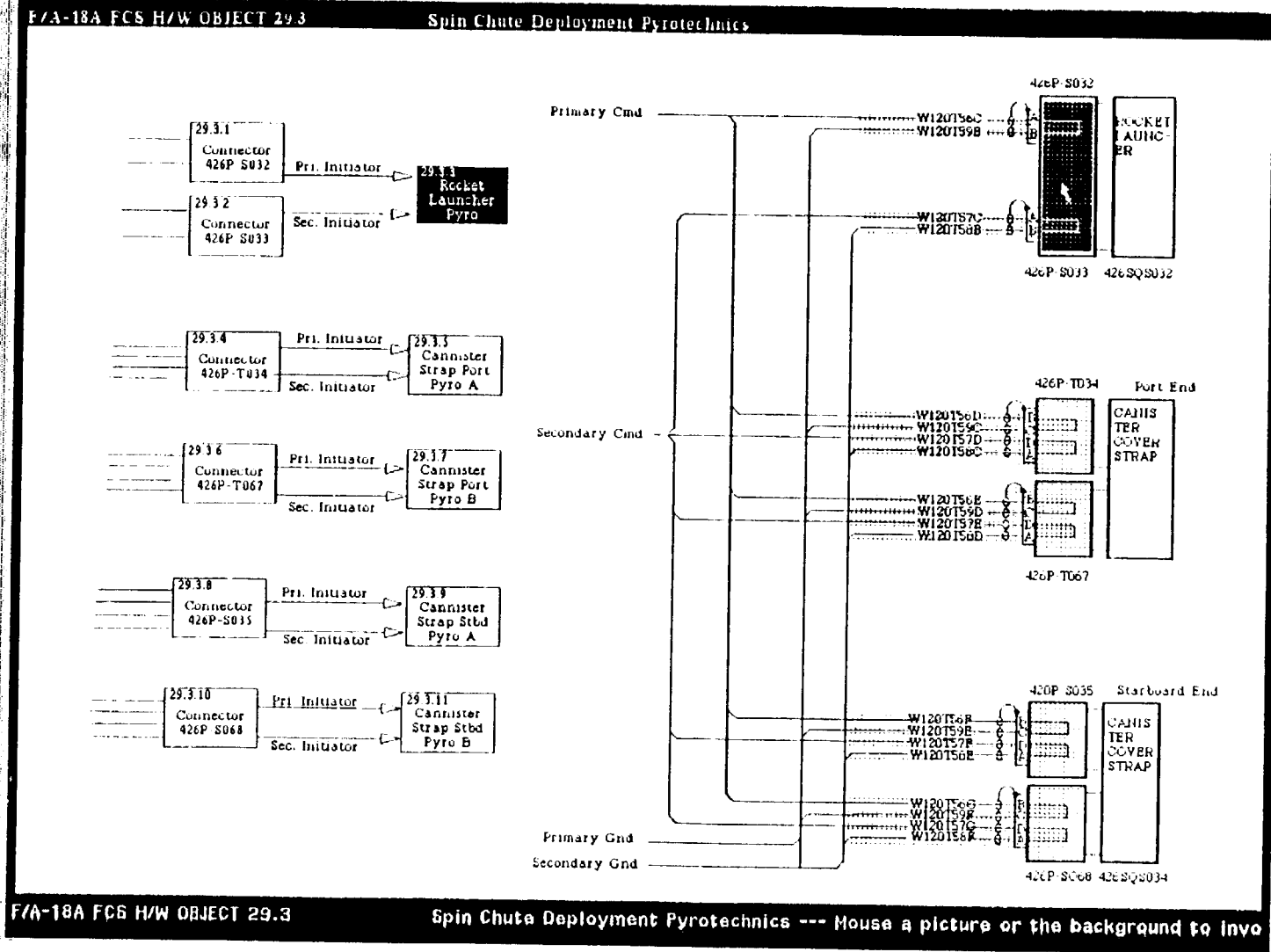


Figure 5-10. Deployment Pyrotechnics H/W Diagram (Level 2)



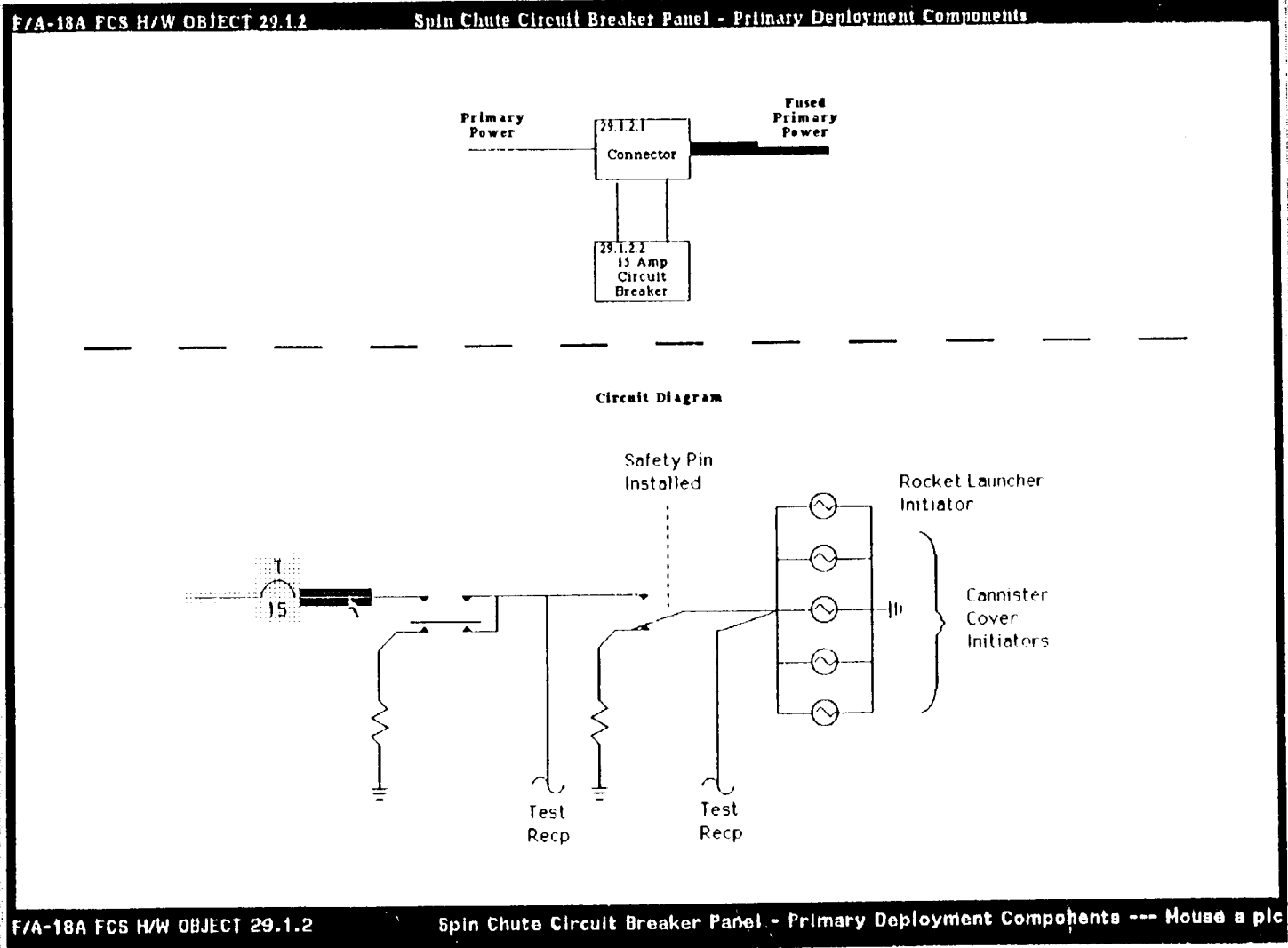


Figure 5-11. Primary Deployment Circuit Breaker Panel Components H/W Diagram (Level 3)

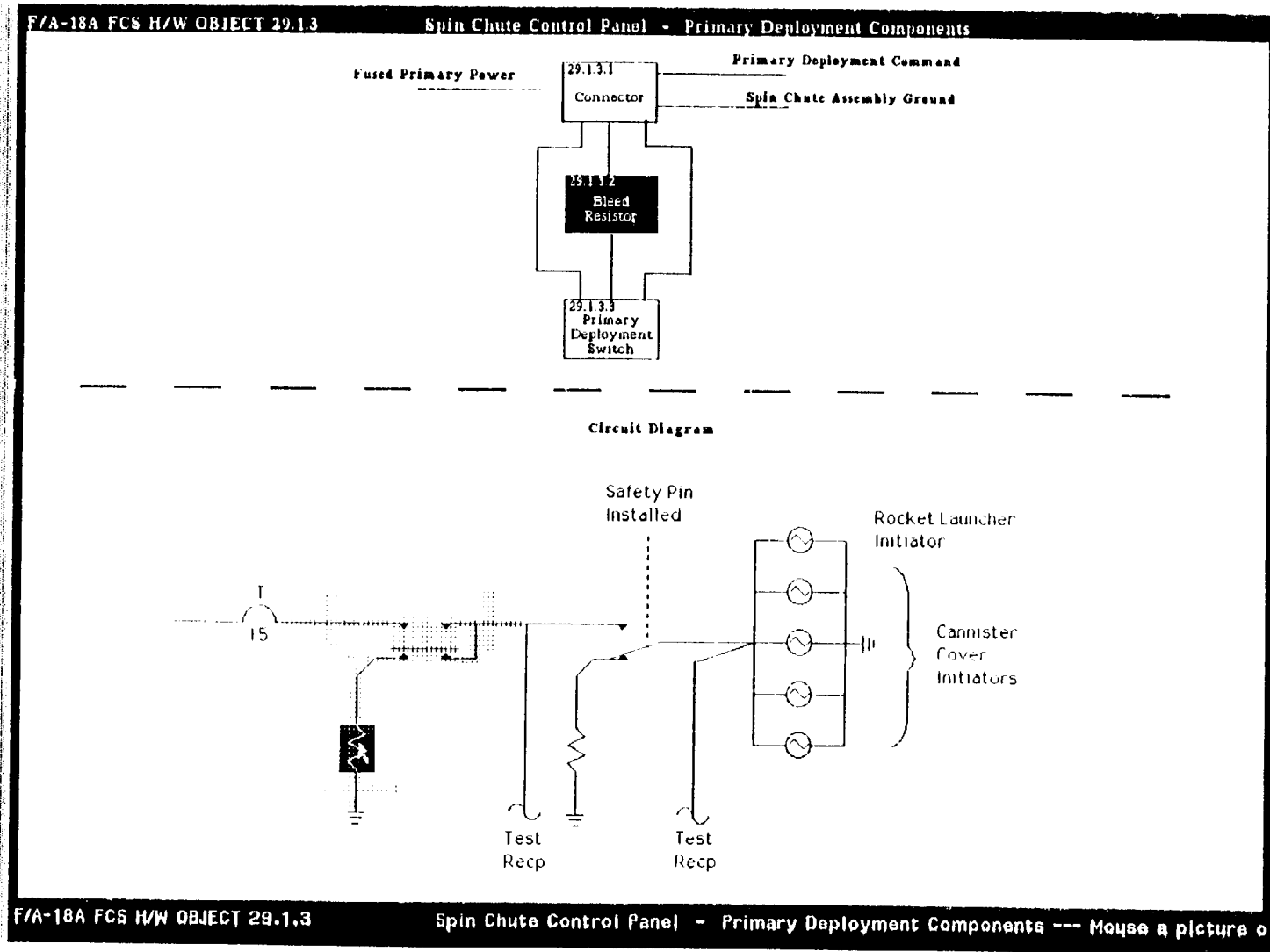


Figure 5-12. Primary Deployment Control Panel Components H/W Diagram (Level 3)

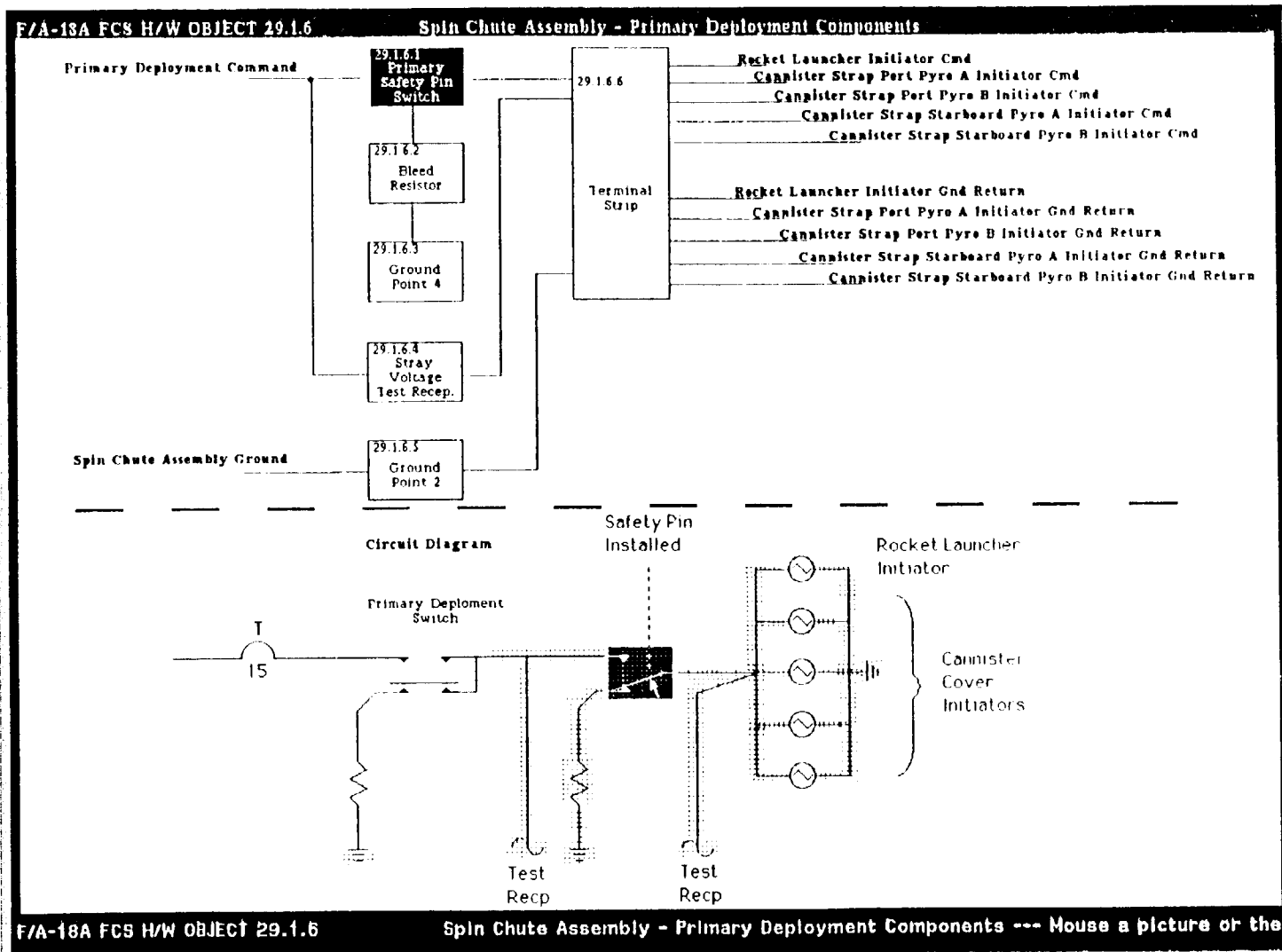


Figure 5-13. Primary Deployment Spin Chute Assembly Components H/W Diagram (Level 3)

### 5.2.2 The Rule-Based Model

A rule-based paradigm is used to model the spin chute deployment functionality. This paradigm utilizes state variables to model the spin chute functionality at the "upper level". For example the state of the cannister is modeled by a variable designated as *cannister.released* (with values of *released* or *not.released*) and this state variable is used in the spin chute deployment rule defined in Figure 5-14. The status of the relevant H/W and S/F objects defined in the H/W diagrams are used to model the components at the "lower level". For example, the *H/W-29.3.3-rocket-launcher-pyro* H/W object defined in Figure 5-10 is used in the primary rocket launch rule defined in Figure 5-18.

The highest level rules, which model the activation of the spin chute, are depicted in Figure 5-14. Spin chute deployment is dependent on the release of a cannister, which holds the chute, and the subsequent launch of a rocket, which deploys the chute. It can be seen that the cannister release and the rocket launch of the spin chute are redundantly activated. Since the rules for the primary and secondary activation are essentially the same, only the rules for the primary activation are focused on here.

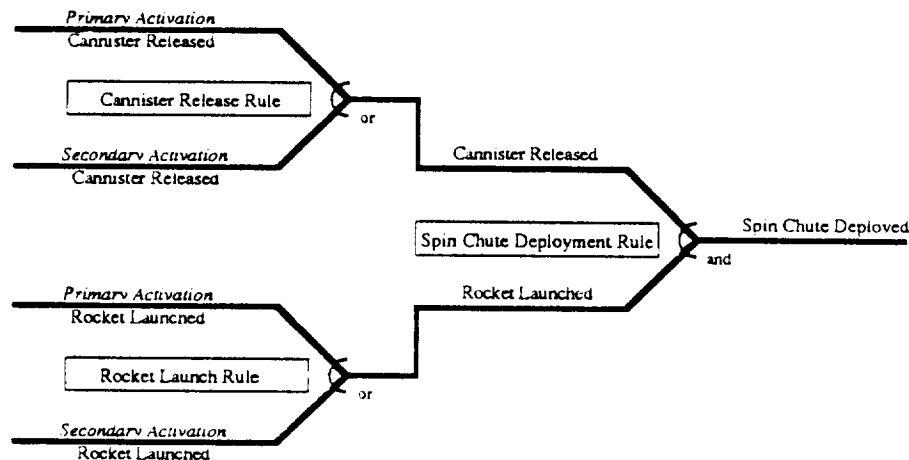


Figure 5-14 - Top Level Chute Deployment Rules.

The primary cannister strap release rules are depicted in Figures 5-16 and 5-17. The primary activation of any one of the four cannister strap pyrotechnics is sufficient to release the cannister strap. The primary rocket launch rules are depicted in Figure 5-18. The

## Rule-Based Models

primary activation of the rocket launch pyrotechnic deploys the spin chute. It can be seen that the primary activation of the rocket launch pyrotechnic is dependent on the presence of a primary signal. The level 2 H/W diagram (Figure 5-8) model of the components which implement the primary rocket initiation are included in the rules depicted in Figure 5-18. The *Dedicated Components OK Rule* defined on the left side of Figure 5-18 is dependent upon the primary rocket initiation components being OK.

The rules which enable the primary signal are depicted in Figures 5-19 and 5-20. The rules in Figure 5-19 focus on the status of the level 2 components (see the H/W diagram in Figure 5-8) and the level 3 command switch positions. The rules in Figure 5-20 focus on the decomposition part status at level 3, where the circuit breaker, control panel and spin chute assembly have been decomposed to attain a finer degree of modeling precision.

### **5.2.3 The FMEA**

The FMEA uses the data driven forward chaining capability of the inference engine. This dynamic model enables the user to selectively fail the spin chute deployment components. The status of the spin chute deployment components can be declared as failed or as operational in the H/W diagrams (see Figures 3.1.1-7 and 3.1.1-8) with mouse activated menu commands. The effect of the selected failures on the spin chute deployment functionality is then modeled by activating the forward chaining paradigm using the rules described previously. The Spin Chute Deployment FMEA user interface (desktop) provides the capability to activate the forward chaining and to display the effects of the selected failures on the spin chute deployment functionality.

The FMEA desktop is depicted in Figure 5-21. This desktop incorporates a deployment control panel and several status display panels. The user is provided with several mechanisms which enable him/her to control the FMEA model. The safety pin can be removed to enable the spin chute system, deployment components can be initialized as being ok and the deployment command can be issued directly from the control panel. The H/W realm select button displays the H/W design realm where the user can use the mouse to selectively fail components.

Once the FMEA model has been initialized the deployment command can be issued to the rule-based model. The component status for the simulation (as defined by the user) is displayed in the upper right hand window. After the FMEA simulation has been executed,

## Rule-Based Models

the status of the deployment system can then be viewed in the displays located in the bottom portion of the desktop. In the particular case depicted in Figure 5-21, the component failures have induced the functional loss of the primary command circuit, a pyrotechnic and an initiator. The effects of these failures are evident in the status displays.

### 5.2.4 The FTA

The FTA uses the goal driven backward chaining capability of the inference engine. This dynamic model enables the user to explore how the functionality for a given capability depends on the H/W components and to determine its redundancy.

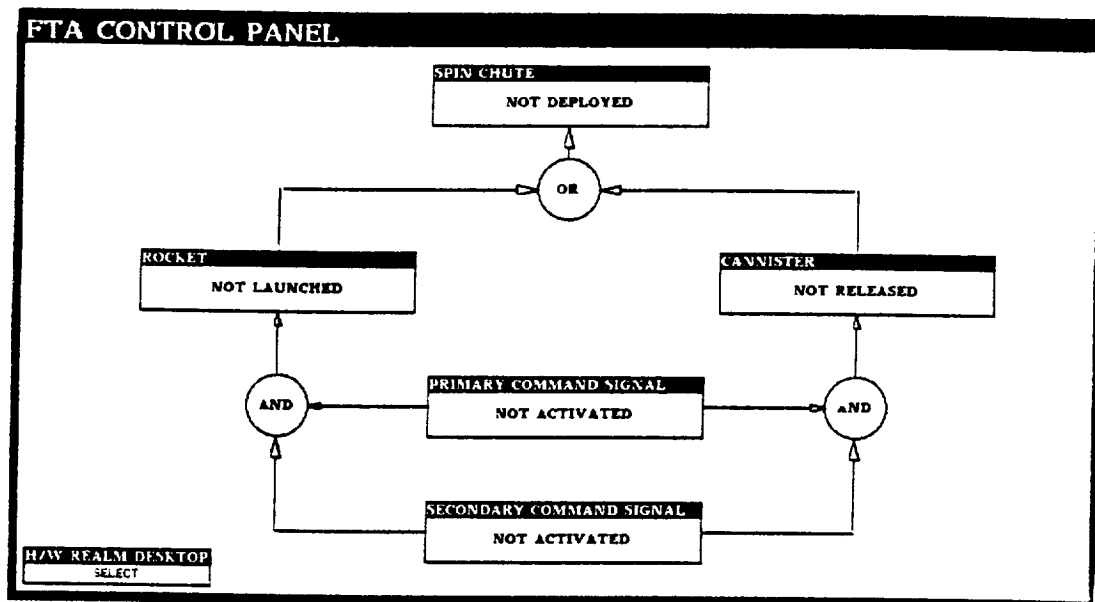


Figure 5-15. FTA Control Panel

The user interface includes a control panel (Figure 5-15) which displays a fault tree for the spin chute deployment. This fault tree depicts the top level logic associated with the failure of the spin chute to deploy. This logic defines the spin chute deployment capability associated with the loss of major subsystem functionality. Each of the functionalities in the FTA Control Panel is mouse sensitive. When selected by the user, the backward chaining paradigm is used to model the selected functionality. The rules described in Section 5.2.2 are used in this model.

## Rule-Based Models

The FTA desktop is depicted in Figure 5-22 with a backward chaining trace of the rules activated when the "SPIN CHUTE - NOT DEPLOYED" box was moused in the FTA Control Panel. The trace in the lower part of the figure indicates that the spin chute is a non-redundant element and is dependent on the release of the cannister followed by the launch of the rocket which pulls the parachute out to the rear of the aircraft. The individual elements of the trace are mouse sensitive and may be used to display the rules used in the backward chaining trace. Here, the spin chute deployment rule was selected for display.

The lower window in Figure 5-23 depicts the backward chaining trace activated when the "ROCKET-NOT LAUNCHED" functionality in the FTA Control Panel is selected by the user. The dual (redundant) solutions for the rocket launch indicate that there are two ways to launch the rocket. The scroll bar in this window allows the user to explore (less than 25% of the trace is shown here) the entire derivation. The individual elements of the backward chaining display are mouse sensitive and allow the user to further explore the rocket launch functionality. Here another form of the trace has been requested and displayed in the upper window. This display focuses on the primary activation of a rocket launch. Figure 5-24 depicts the right half of this primary actuation logic, which indicates the associated components.

The FTA desktop is also depicted in Figure 5-25 with a backward chaining trace of the rules activated when the "CANNISTER - NOT RELEASED" box was moused in the FTA Control Panel. The trace indicates that there are 8 ways in which the cannister release can be effected. These multiple logical traces for the release of the cannister answer the question "*How redundant is the cannister release implementation?*". This 8 fold redundancy reflects the ability to actuate the four individual cannister strap pyrotechnics with either the primary or secondary command signals. The necessary individual components can be found in the scrollable window which displays the 8 redundant cannister release mechanizations. Inspection of the individual cannister release backward traces allows the user to answer the question "*What components must work for this cannister release capability?*". In Figure 5-26, the cannister release backward chaining trace has been scrolled to depict the goal end of the trace.

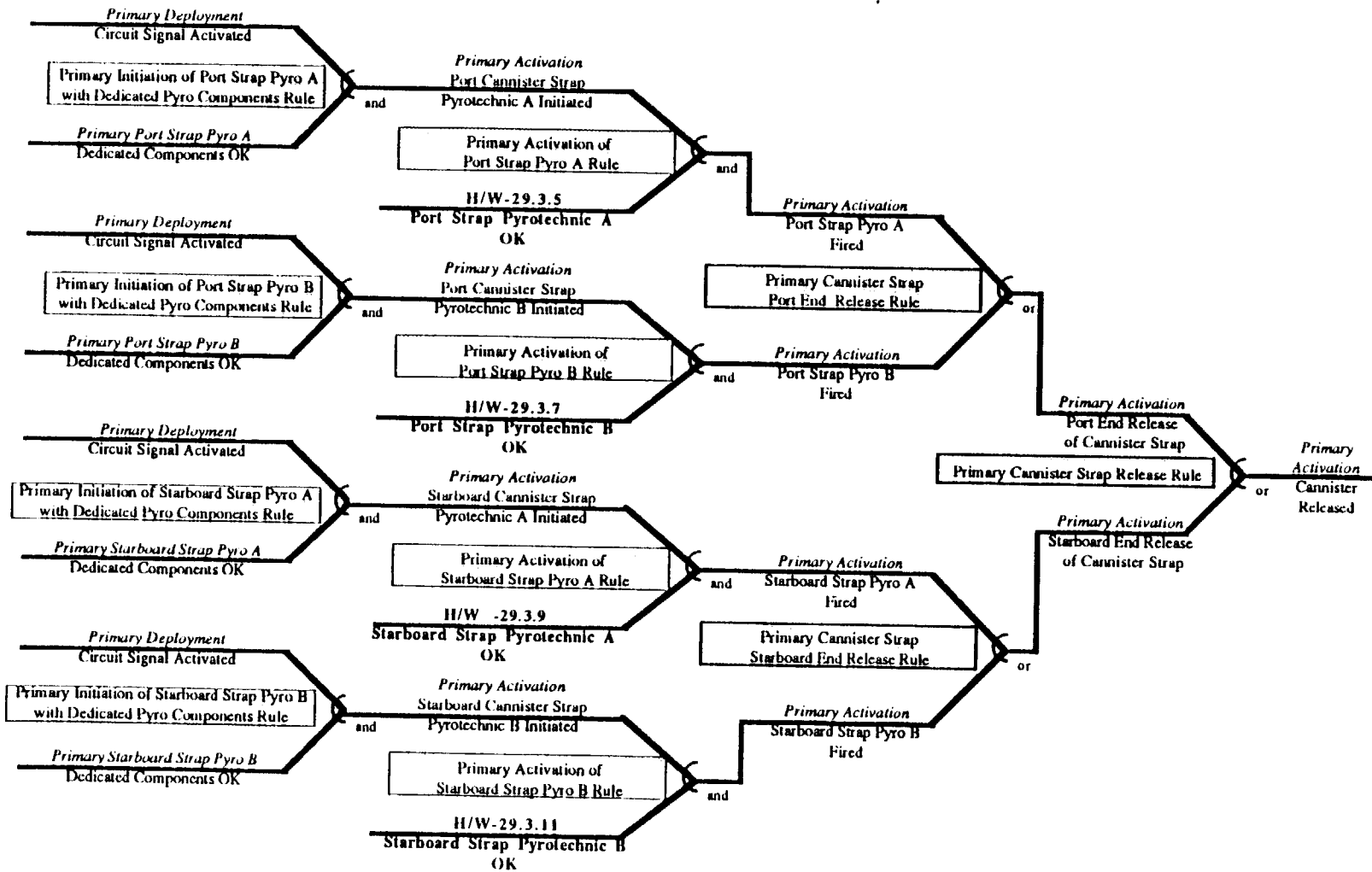


Figure 5-16. Primary Cannister Strap Release Rules - Part I (Release Logic).



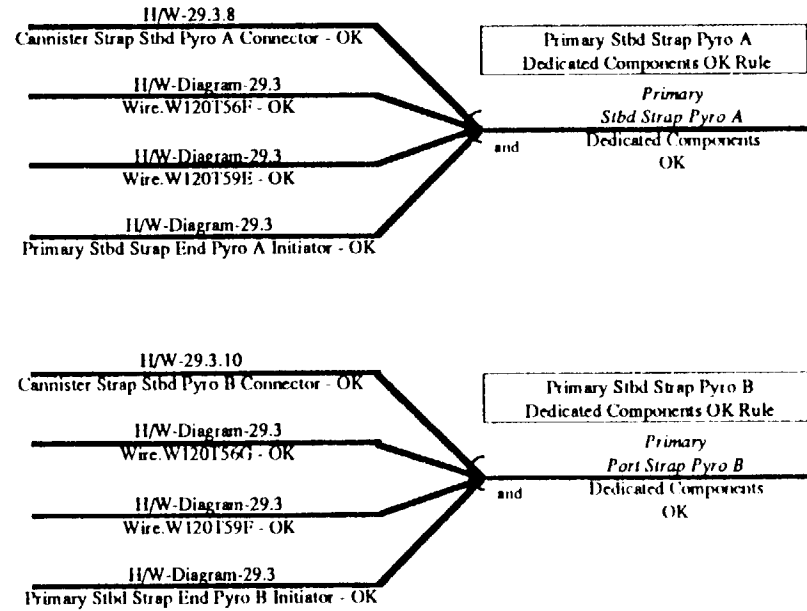
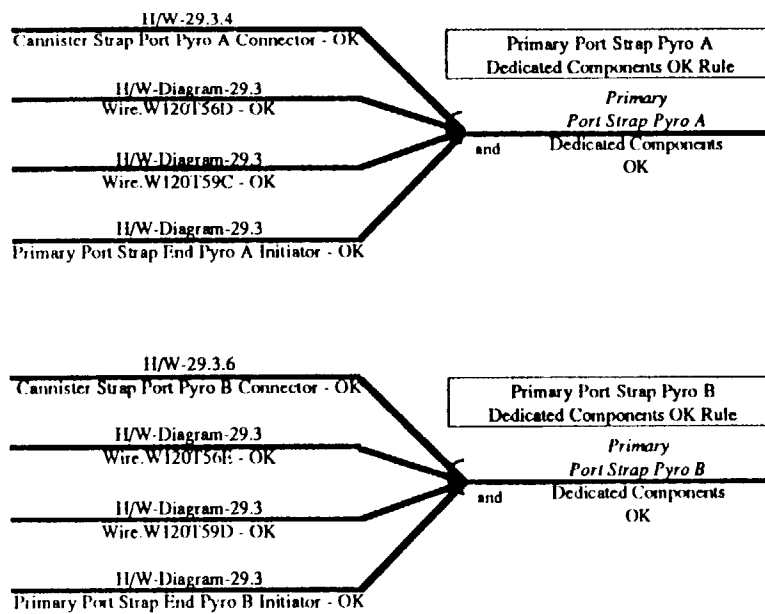


Figure 5-17. Primary Cannister Strap Release Rules - Part II (Pyrotechnic Components).

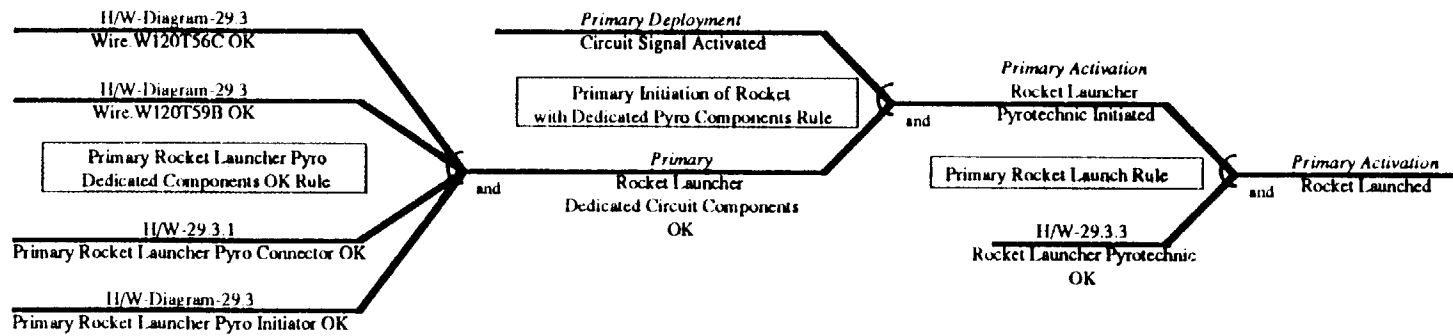


Figure 5-18. Primary Rocket Launch Rules

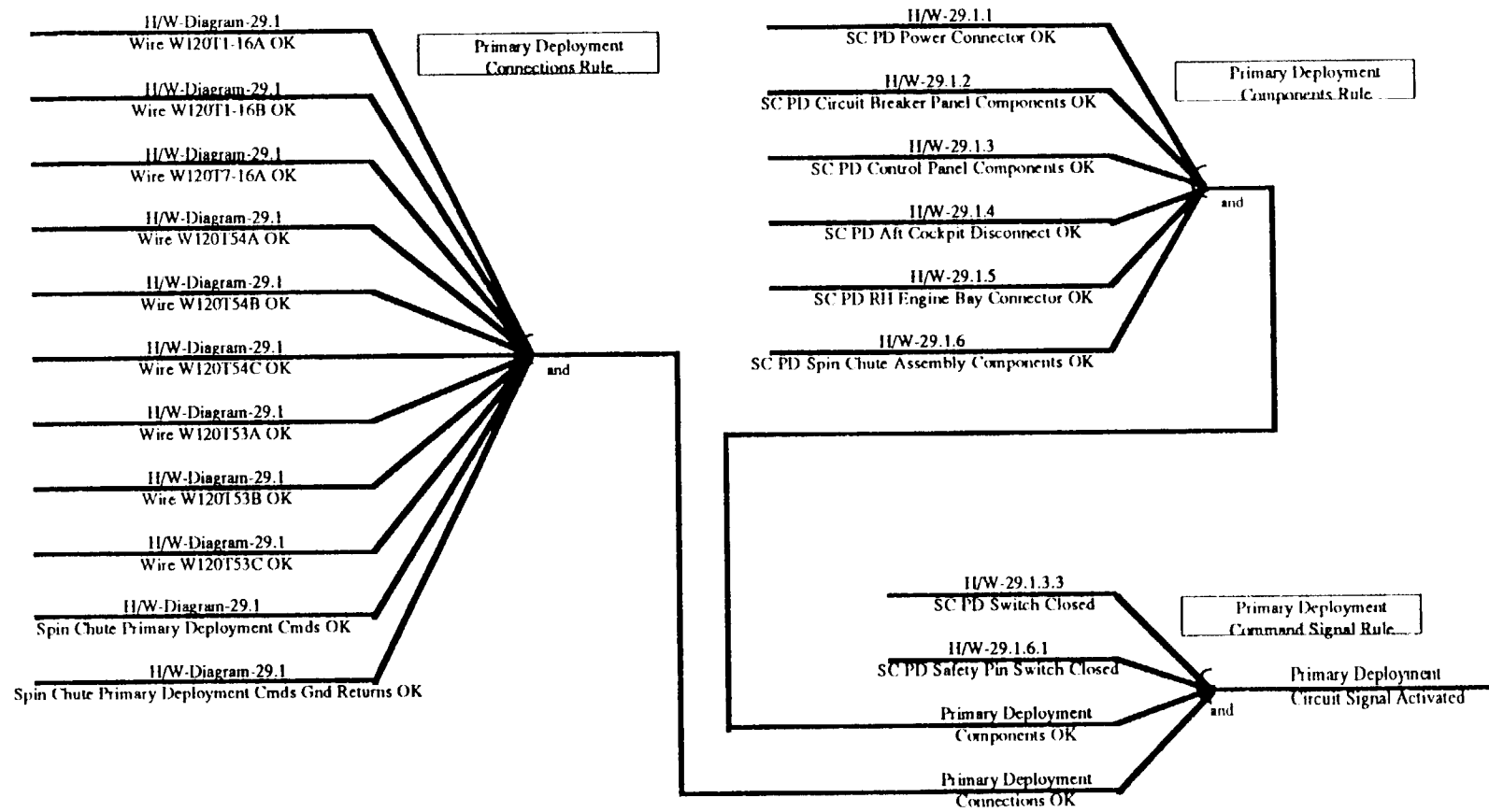


Figure 5-19. Primary Deployment Circuit Activation Rules - Part I (Level 2 Components & Level 3 Switch Cmds).

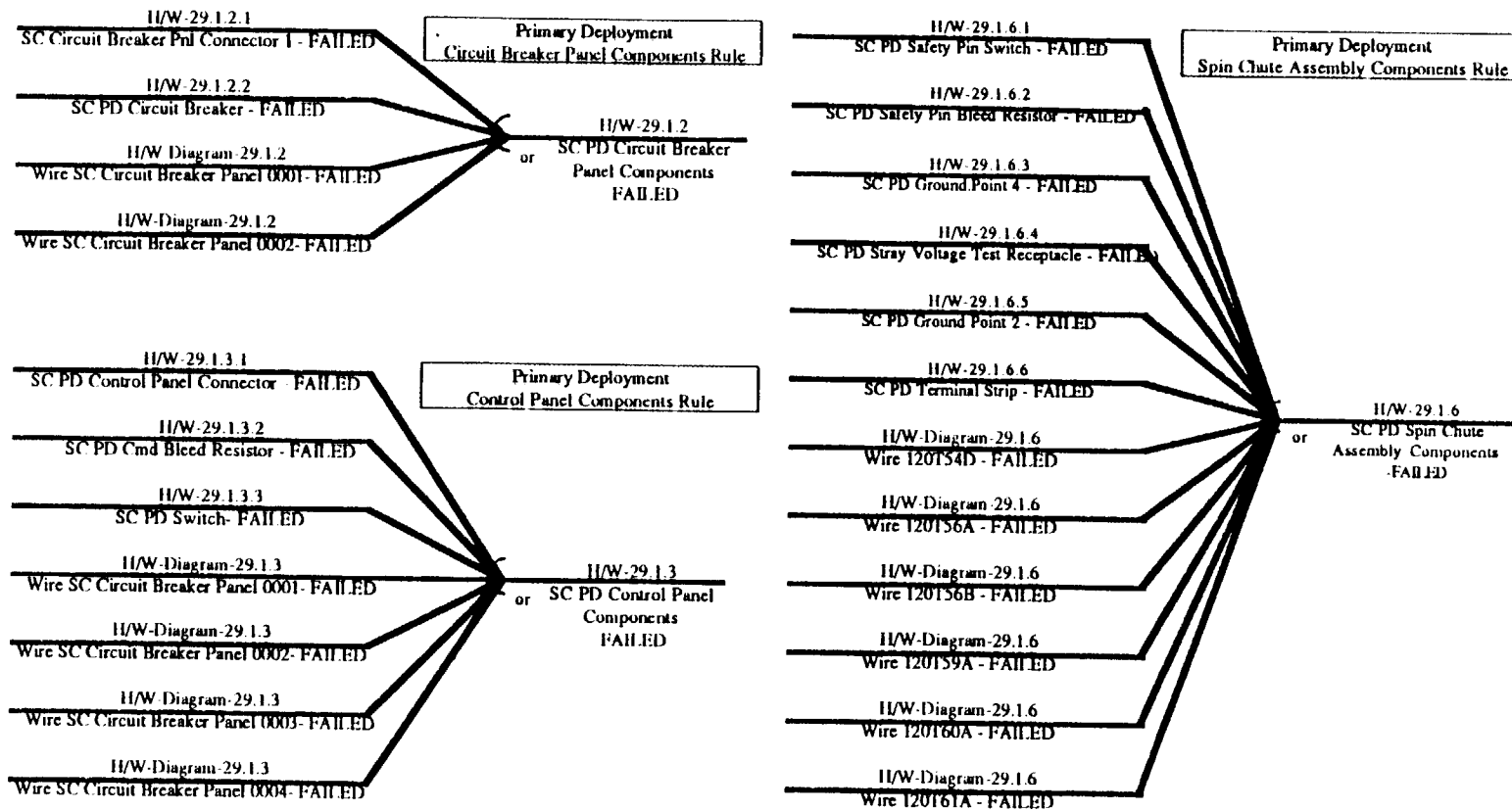





Figure 5-20. Primary Deployment Circuit Activation Rules - Part II (Level 3 Components).

**Spin Chute Deployment - FMEA Model**

### FMEA CONTROL PANEL

**DEPLOYMENT COMMAND**  
DEPLOY SPIN CHUTE

**FMEA INITIALIZATION**  
INITIALIZE

**H/W REALM DESKTOP**  
SELECT

**SAFETY PIN**  
IN PLACE ☒

### STATUS OF SPIN CHUTE DEPLOYMENT COMPONENTS

The Status of the Spin Chute Deployment Components IS NOW BEING CHECKED.  
..... (Only FAILED components are reported.)

- > H/W-29.1.1-SC.PRI.DEP.POWER.CONNECTOR is FAILED.
- > H/W-29.3.5-CANNISTER.STRAP.PORT.PYRO.A is FAILED.
- > H/W-DIAGRAM-29.3-SEC.STBD.STRAP.END.PYRO.A.INITIATOR is FAILED.

The Status of the Spin Chute Deployment Components HAS NOW BEEN CHECKED.  
..... (If no failures are indicated all of the components are operational.)

### DEPLOYMENT STATUS

**CANNISTER**  
NOT RELEASED ☒ RELEASED

**ROCKET**  
NOT LAUNCHED ☒ LAUNCHED

**SPIN CHUTE**  
NOT DEPLOYED ☒ DEPLOYED

### PYROTECHNIC STATUS

--- Primary Initiation ---
--- Secondary Initiation ---
--- Pyro Status ---

<b>Port Cannister Strap Pyrotechnics</b>		
<b>PRI PORT PYRO A</b> <input checked="" type="checkbox"/> INITIATED	<b>SEC PORT PYRO A</b> NOT INITIATED <input checked="" type="checkbox"/> INITIATED	<b>PORT STRAP PYROTECHNIC A</b> NOT FIRED <input checked="" type="checkbox"/> FIRED
<b>PRI PORT PYRO B</b> <input checked="" type="checkbox"/> INITIATED	<b>SEC PORT PYRO B</b> NOT INITIATED <input checked="" type="checkbox"/> INITIATED	<b>PORT STRAP PYROTECHNIC B</b> NOT FIRED <input checked="" type="checkbox"/> FIRED
<b>Starboard Cannister Strap Pyrotechnics</b>		
<b>PRI STBD PYRO A</b> <input checked="" type="checkbox"/> INITIATED	<b>SEC STBD PYRO A</b> <input checked="" type="checkbox"/> INITIATED	<b>STBD STRAP PYROTECHNIC A</b> NOT FIRED <input checked="" type="checkbox"/> FIRED
<b>PRI STBD PYRO B</b> <input checked="" type="checkbox"/> INITIATED	<b>SEC STBD PYRO B</b> NOT INITIATED <input checked="" type="checkbox"/> INITIATED	<b>STBD STRAP PYROTECHNIC B</b> NOT FIRED <input checked="" type="checkbox"/> FIRED
<b>Rocket Launcher Pyrotechnic</b>		
<b>PRI ROCKET</b> <input checked="" type="checkbox"/> INITIATED	<b>SEC ROCKET</b> NOT INITIATED <input checked="" type="checkbox"/> INITIATED	<b>ROCKET PYROTECHNIC</b> NOT FIRED <input checked="" type="checkbox"/> FIRED

### COMMAND STATUS

**PRIMARY COMMAND SIGNAL**  
NOT ACTIVATED ☒ ACTIVATED

**SECONDARY COMMAND SIGNAL**  
NOT ACTIVATED ☒ ACTIVATED

The FMEA RULES HAVE BEEN RUN.

Figure 5-21. Spin Chute Deployment Model FMEA Interface.

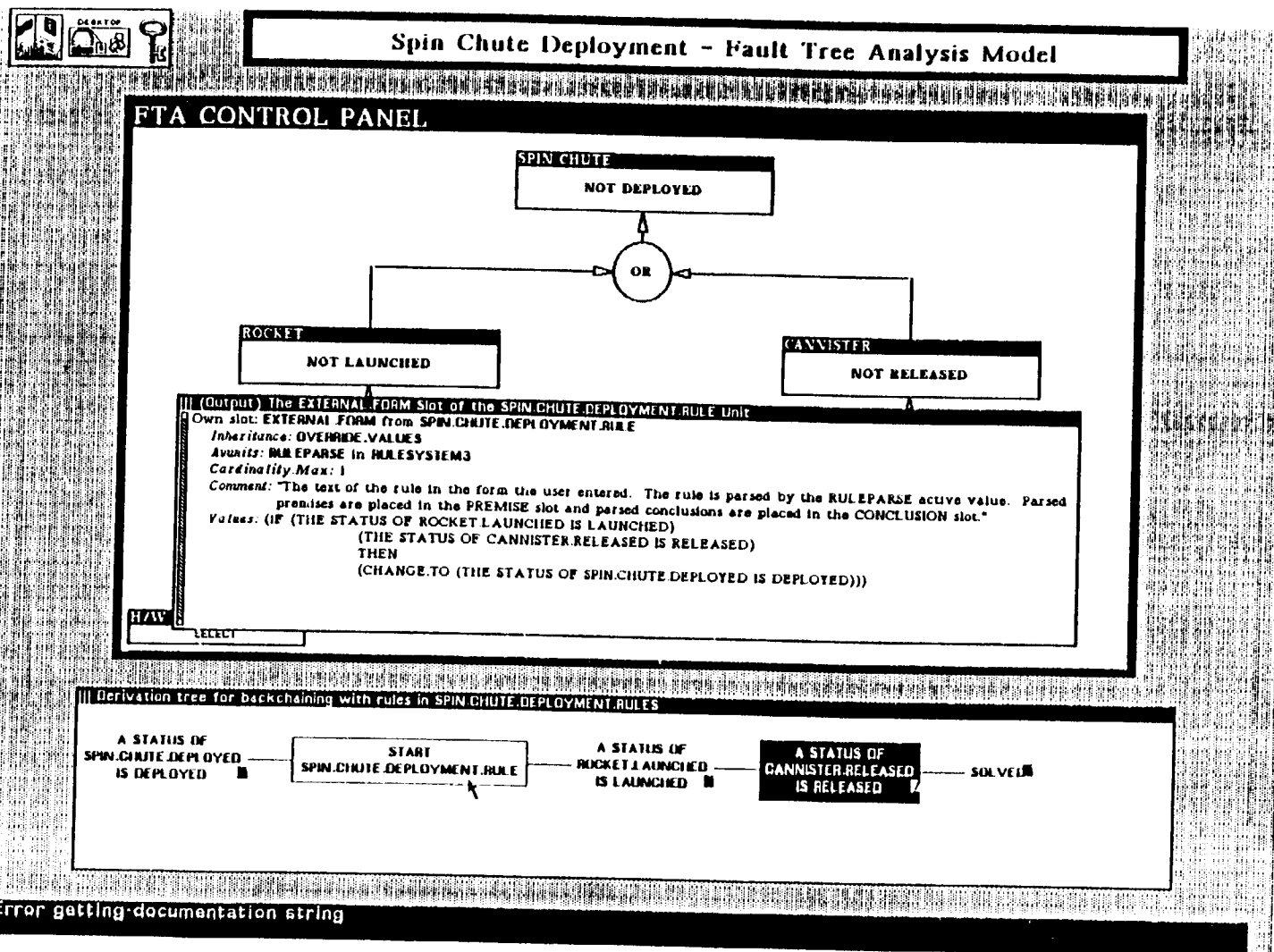


Figure 5-22. Spin Chute Deployment FTA Rule-Based Diagnosis.

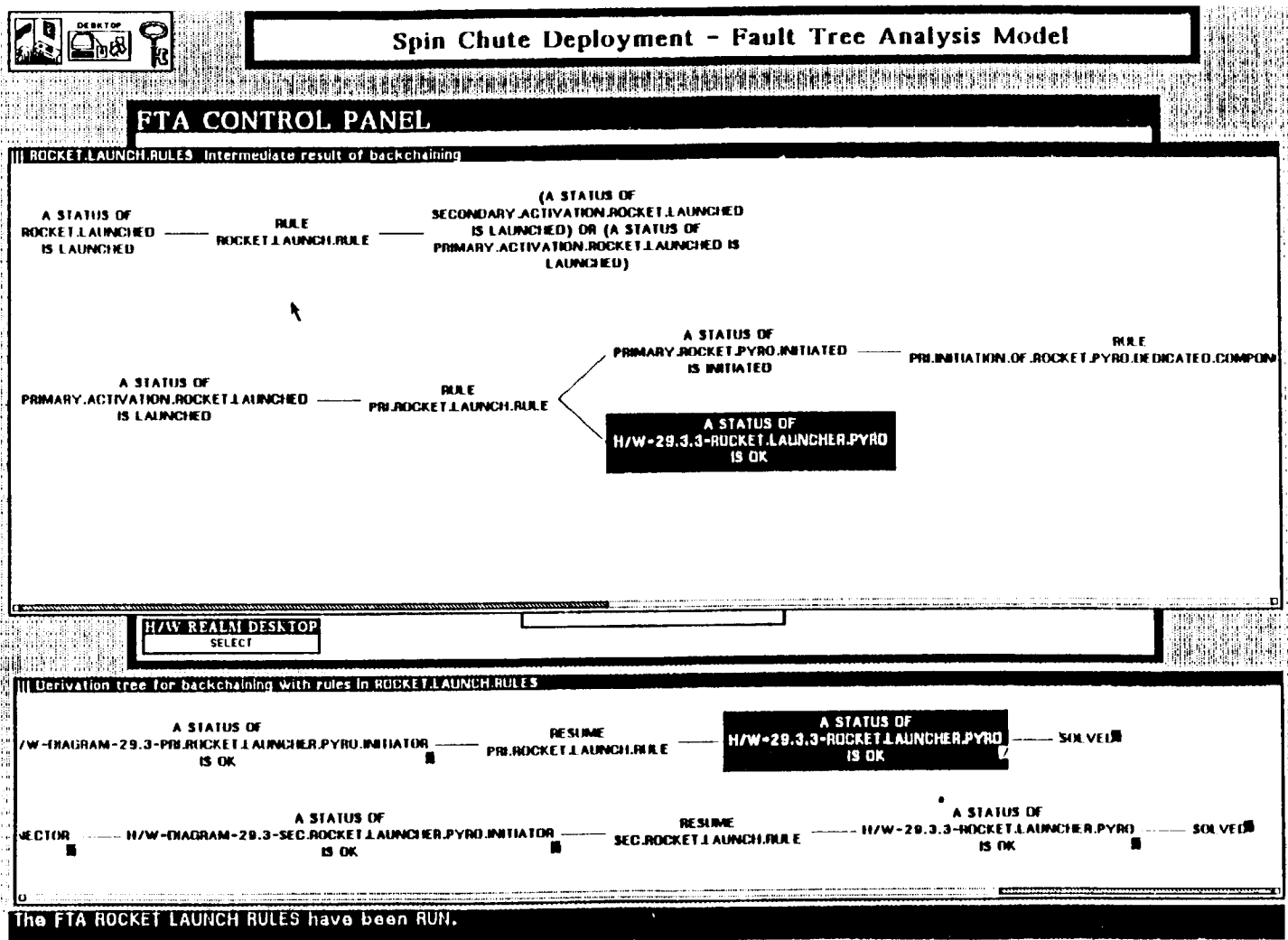


Figure 5-23. Spin Chute Rocket Launch FTA Rule-Based Diagnosis - Display #1.

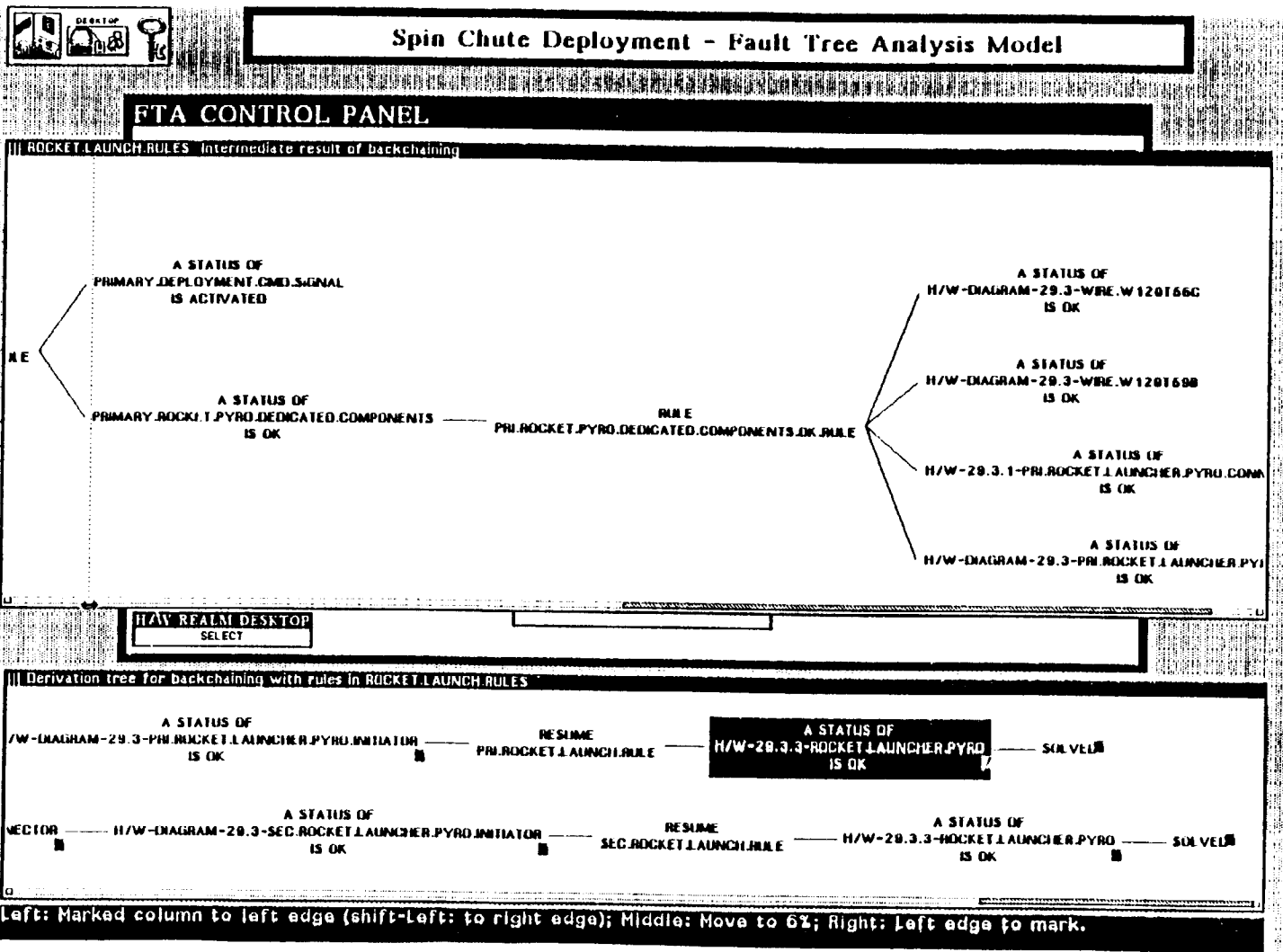


Figure 5-24. Spin Chute Rocket Launch FTA Rule-Based Diagnosis - Display #2.



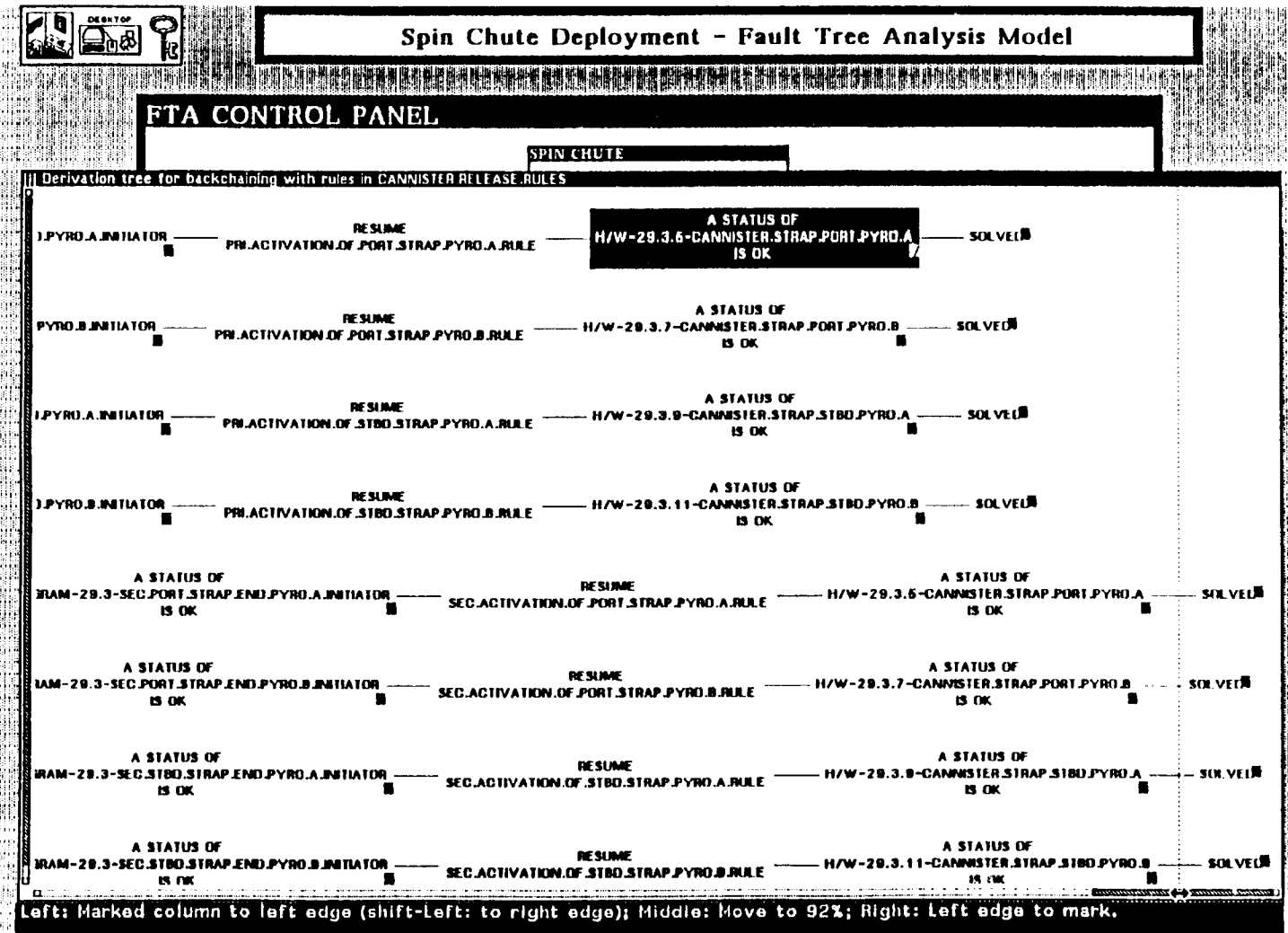


Figure 5-25. Spin Chute Cannister Release FTA Rule-Based Diagnosis - Display #1.

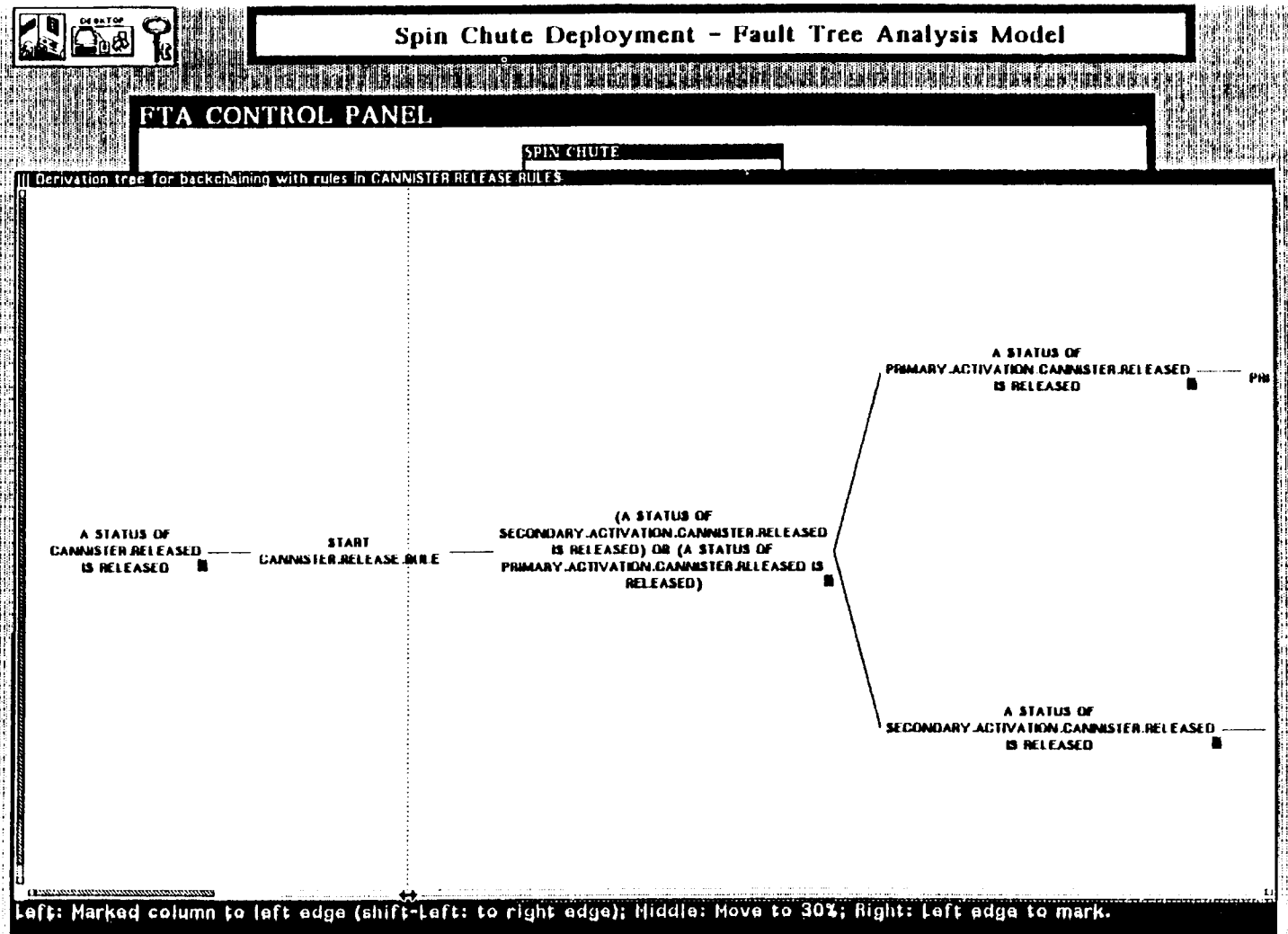


Figure 5-26. Spin Chute Cannister Release FTA Rule-Based Diagnosis - Display #2.

### 5.2.5 Conclusions

The spin chute deployment model successfully links the structural design knowledge captured by the top down decomposition mechanisms with the behavioral knowledge displayed by the rule-based model. This integration of the structural and behavioral knowledge gives the designer a better understanding of the functionality of his/her design and also provides useful information to the FCS users later on in the FCS life cycle. Here, the life cycle is seen to range from design phase, through test and operation and on into the retrofit phase.

It is important to recognize that this captured design knowledge is useful throughout the FCS life cycle, because considerable resources are required to capture this knowledge. A recognition that these costs can be amortized over the entire life cycle is considered to be necessary, if the knowledge capture investment is to be justified up front in the design phase.

In this case, the behavioral model was defined by a set of and/or trees which specify behavior in terms of the objects captured/defined in the H/W design realm. These and/or trees possess the positive quality of communicating well to both the application community and the knowledge engineer. Once this form of behavior specification satisfies the application community, the knowledge engineer can define a set of rules in a format that is recognized by an inference engine. This *single* set of rules can then be used to run a dynamic set of FTAs and FMEAs interactively which allow the user to explore *many* different failure implications in real time. This ability to obtain many different analyses from a single model is viewed as a powerful design capability.

Using 20/20 hindsight on this model it is seen that the KEE™ forward and backward chaining graphics are not exactly what is desired to support FMEA and FTA displays. The modeling techniques discussed in Section 5.3 dealt with this issue and developed a set of graphics which are more nearly what is desired to support these safety analyses. Another area that deserves additional attention deals with the and/or trees. The development of these trees was time consuming. The development of interactive automated design aids, which would assist in the development of behavioral specifications using and/or trees, would certainly have simplified the development of the spin chute deployment model. Such an interactive design aid should be considered for a Phase 2 effort.

## Rule-Based Models

### **5.3 Emergency Hydraulic System Model**

The emergency hydraulic system (EHS) was selected to exercise the schematic capture mechanisms, rule definition aids and behavior modeling mechanisms developed in this project. This life critical, replicated control system was selected to provide a testing ground for these mechanisms which are intended to enhance the confidence that we place in our automated systems. It should be recognized from the outset that this model is only a small step beyond the simple demonstration motor control system discussed earlier in Section 3.0. In fact, the actual EHS has been simplified to comply with resource constraints. A simplified version of the EHS is depicted in Figure 5-27. This version of the EHS upgrades the complexity of the motor control example principally by introducing a replicated power supply. The model discussed in this section was developed as part of a thesis<sup>7</sup> by Scott Sikora.

#### **5.3.1 The H/W Bottom Up Aggregation**

The EHS model utilizes the emergency system schematics to obtain the design specification. These schematics are depicted in Figures 4.4-3 through 4.4-11. The schematic capture mechanisms were used to specify the model abstraction taken from these schematics. Figures 4.4-8 through 4.4-11 are illustrated at a scale factor that permits individual elements of the schematic associated with the EHS to be identified. These elements are highlighted. The schematic capture process was used to develop the bottom level abstraction of the EHS depicted in Figure 5-28. This bottom level abstraction was aggregated twice such that there are three abstractions for the EHS model. Figure 5-28, which depicts the bottom level abstraction contains the most elements and is designated as level 2. The first simplification in the bottom up aggregation process is depicted in Figure 5-29 and is designated as level 1. The next simplification is depicted in Figure 5-30.

These bottom up aggregation diagrams typify a bottom up hierarchy. They provide the typical insight obtained by grouping the elements of a basic design so as to form the functional objects at a higher level. The menu driven authoring mechanisms aid the user in forming these bottom up groupings in the design process and capture the design knowledge in a bottom up fashion. Other menu driven browsing mechanisms are then available for browsing this design at a later time both for testing and for tutoring purposes.

## Rule-Based Models

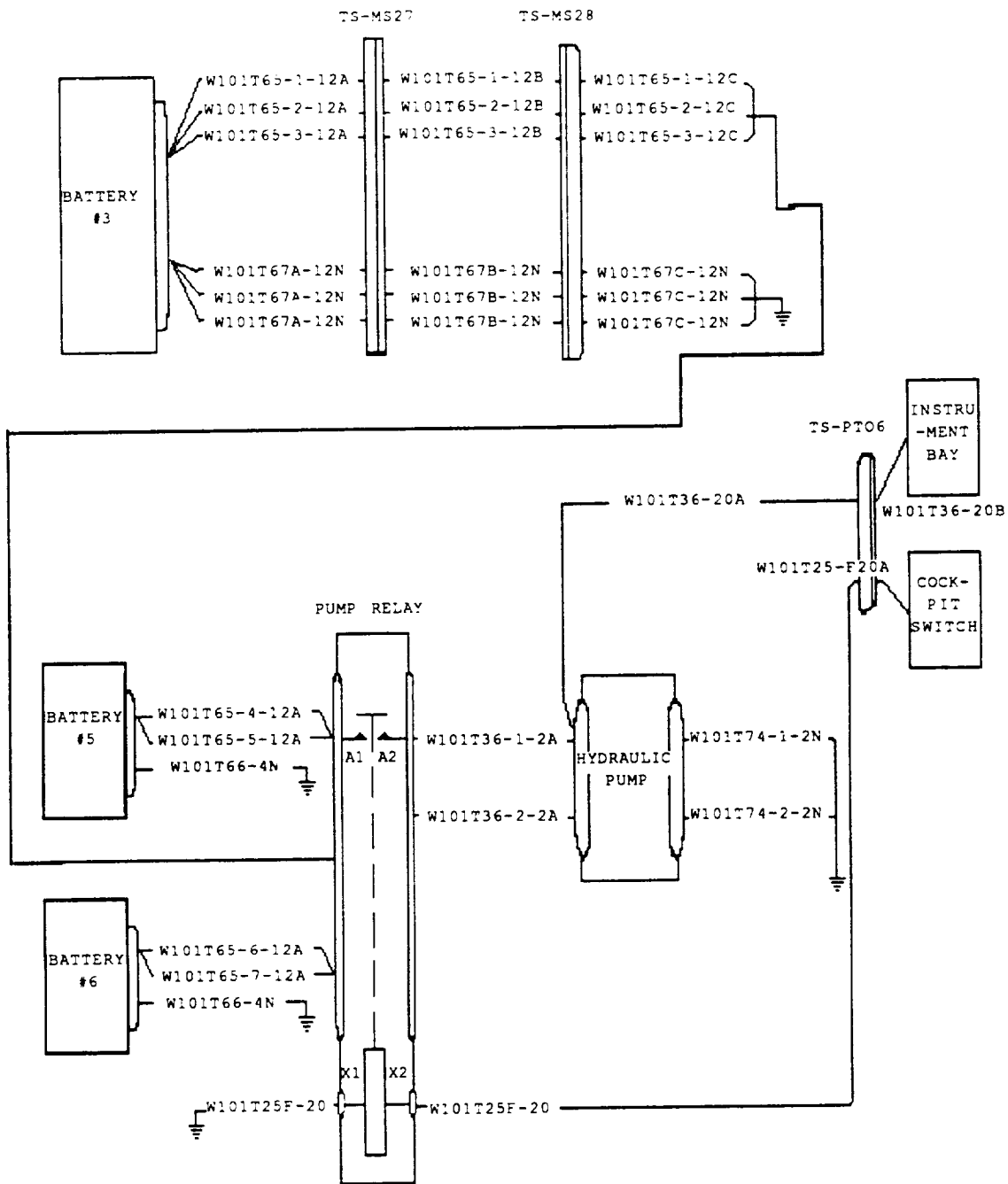


Figure 5-27. A Simplified HARV Emergency Hydraulic System Schematic

## Rule-Based Models

### **5.3.2 The Rule-Based Model**

The rule based model is based upon the bottom abstraction (level 2). The rule capture mechanisms were used to define the behavior for each object in the bottom level abstraction depicted in Figure 5-28. Figure 5-31 depicts a typical rule capture. Here, a menu of options is provided for the user to assist in defining the behavior when a ground connection is failed. It should be noted that the rule generation is not a totally automated process. Rather a set of mechanism have been provided that help the user to specify the behavior. In fact it is not envisioned here that total automation is possible.

In keeping with the philosophy that the user must define the rules in an interactive fashion, a display mechanism is provided for the abstract objects. The rules for one of the many objects in the bottom abstraction are depicted in Figure 5-32. Here, the rules associated with the behavior of a terminal strip are displayed. The rule display option is found on the menu provided for objects in the abstract diagrams. This rule display mechanism allows the user to browse the behavior which has been defined for the EHS model.

This model has focused on the generation of the rules for the bottom level abstraction. It should be noted that rules and behavior for the level 1 and level 0 aggregated views of the EHS could also have been developed.

### **5.3.3 The FMEA**

Once the rules for the objects depicted in Figure 5-3.2 have been defined it is possible to utilize the inference engine to display a *Failure Mode and Effects Analysis* for any of the objects in this bottom level abstraction. Figure 5-33 depicts a typical FMEA diagram, which in this case is for the hydraulic pump relay. This diagram was generated in real time by simply mousing on the hydraulic relay pump symbol (PMP-RLY-1) depicted in Figure 5-28 and selecting the *Failure Mode and Effects Analysis* item on the menu.

### **5.3.4 The FTA**

A *Fault Tree Analysis* for an emergency system hydraulic system pump is illustrated in Figures 5-34, through 5-36. An overview of the FTA is given the Figure 5-34, which

## Rule-Based Models

indicates the scope the analysis. Scrolling and zooming mechanisms allow the user to explore this FTA in more detail. Figure 5-35. zooms in on the top level of this FTA. Figure 5-36. scrolls to another area of the FTA and displays a positioning mechanisms which helps the user browse around this large diagram. Here again, this diagram was generated in real time by simply mousing on the hydraulic pump (PMP-1) depicted in Figure 5-28 and selecting the Fault Tree Analysis item on the menu

### **5.3.5 Conclusions**

The EHS model indicates that the symbolic processing mechanisms developed to support interactive safety analyses during the design process are viable for the restricted model treated here. These mechanisms support schematic capture, behavior rule definition, FMEAs and FTAs in a real time interactive design environment. Furthermore, the interactive mechanisms provide a friendly mouse and menu user interface with a graphical interface which is tailored to the application. The successful results of this modeling effort encourage follow on efforts which focus on enhancing the mechanisms and scaling up the magnitude of the model. A primary area of improvement lies in coordinating an improved definition of the object failure modes with an application engineer. These improvements should also include enhancements to the existing library used to support the rule definition mechanisms.

The bottom up design capture effort was specifically directed toward performance improvement in the area of safety analysis models. Here, a powerful combination of mechanisms for schematic capture and rule capture have been developed which facilitate the interactive development of FMEAs and FTAs and the EHS model demonstrates their use. These design aids significantly enhance the ability of the designer to perform safety analyses. It is likely that this enhanced analysis capability can be used to significantly increase the safety margins and fault tolerance seen in a final design.

Bottom-up expansion of Simplified Emergency Hydraulic System -- Abstract Level 2

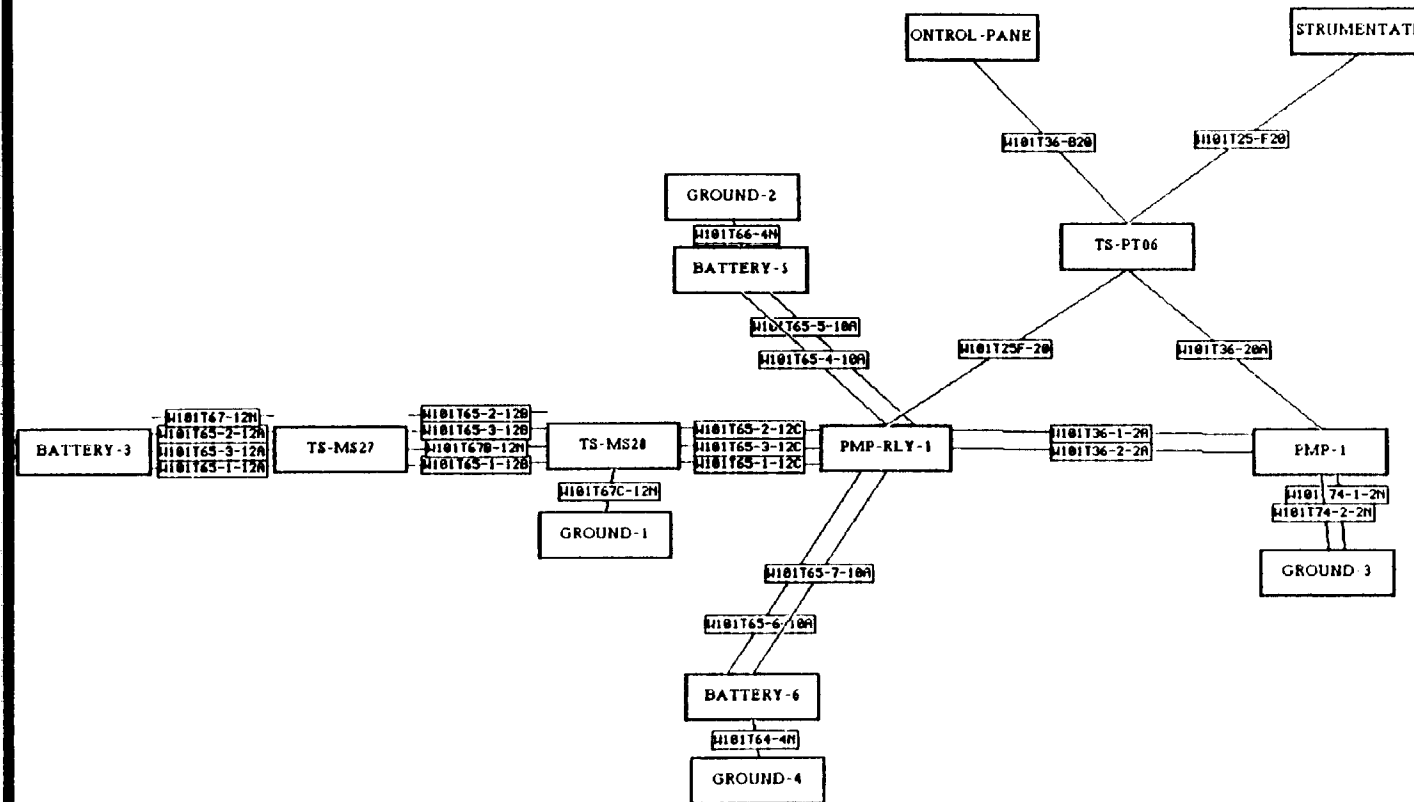


Figure 5-28. Bottom Level Abstraction Diagram for Emergency Hydraulic System Model - Level 2



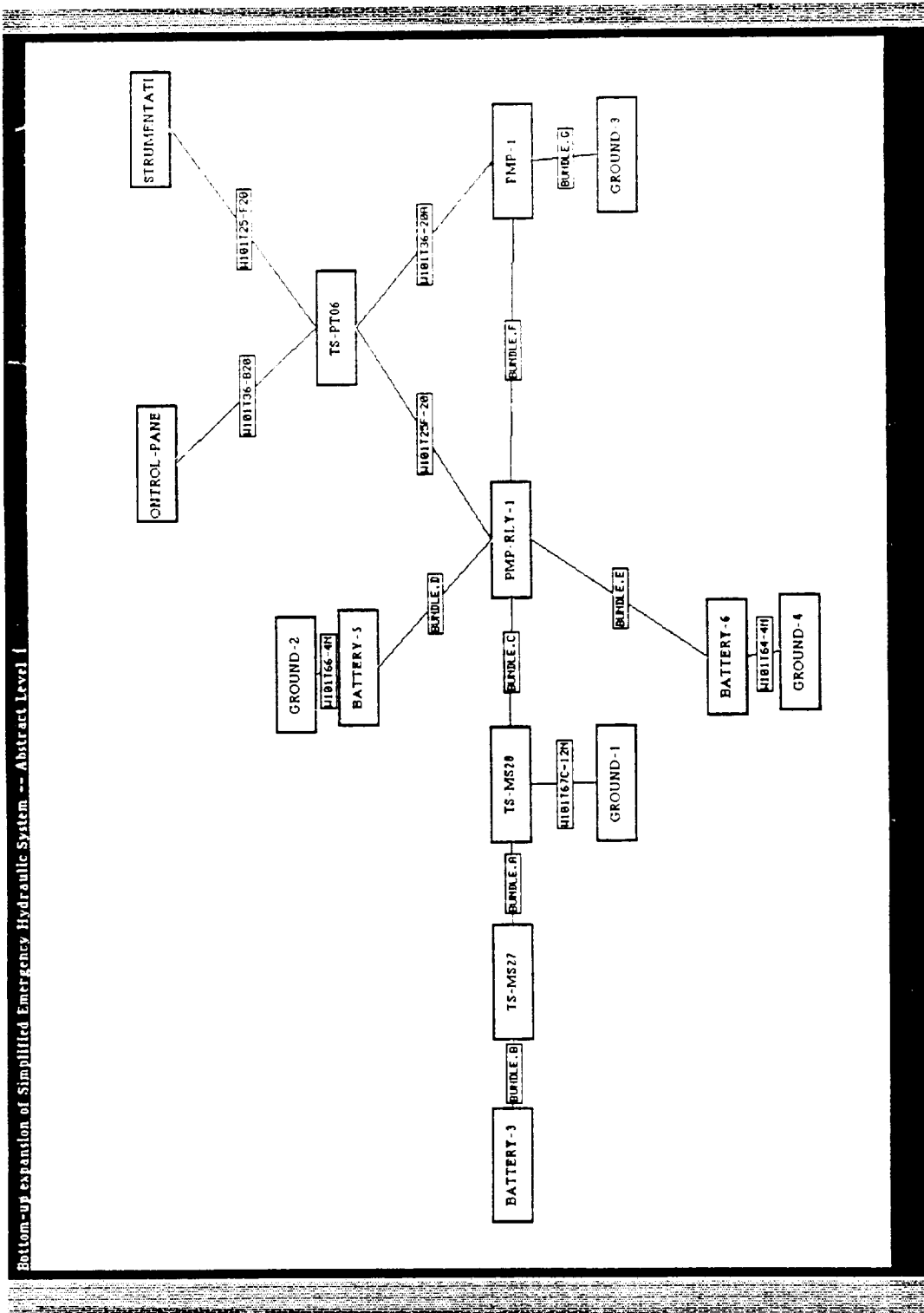


Figure 5-29. First Bottom Up Abstraction Diagram for Emergency Hydraulic System Model - Level 1

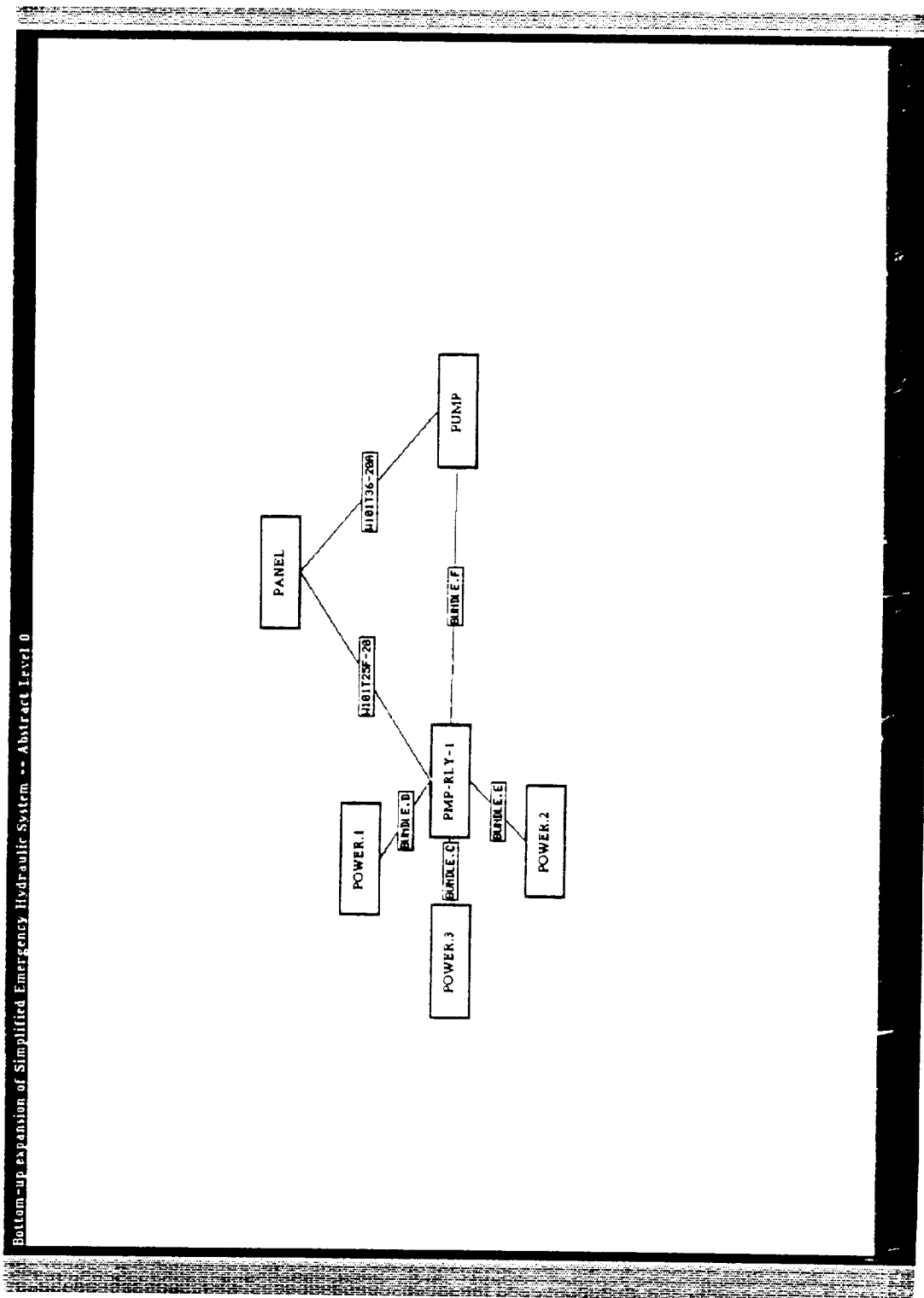
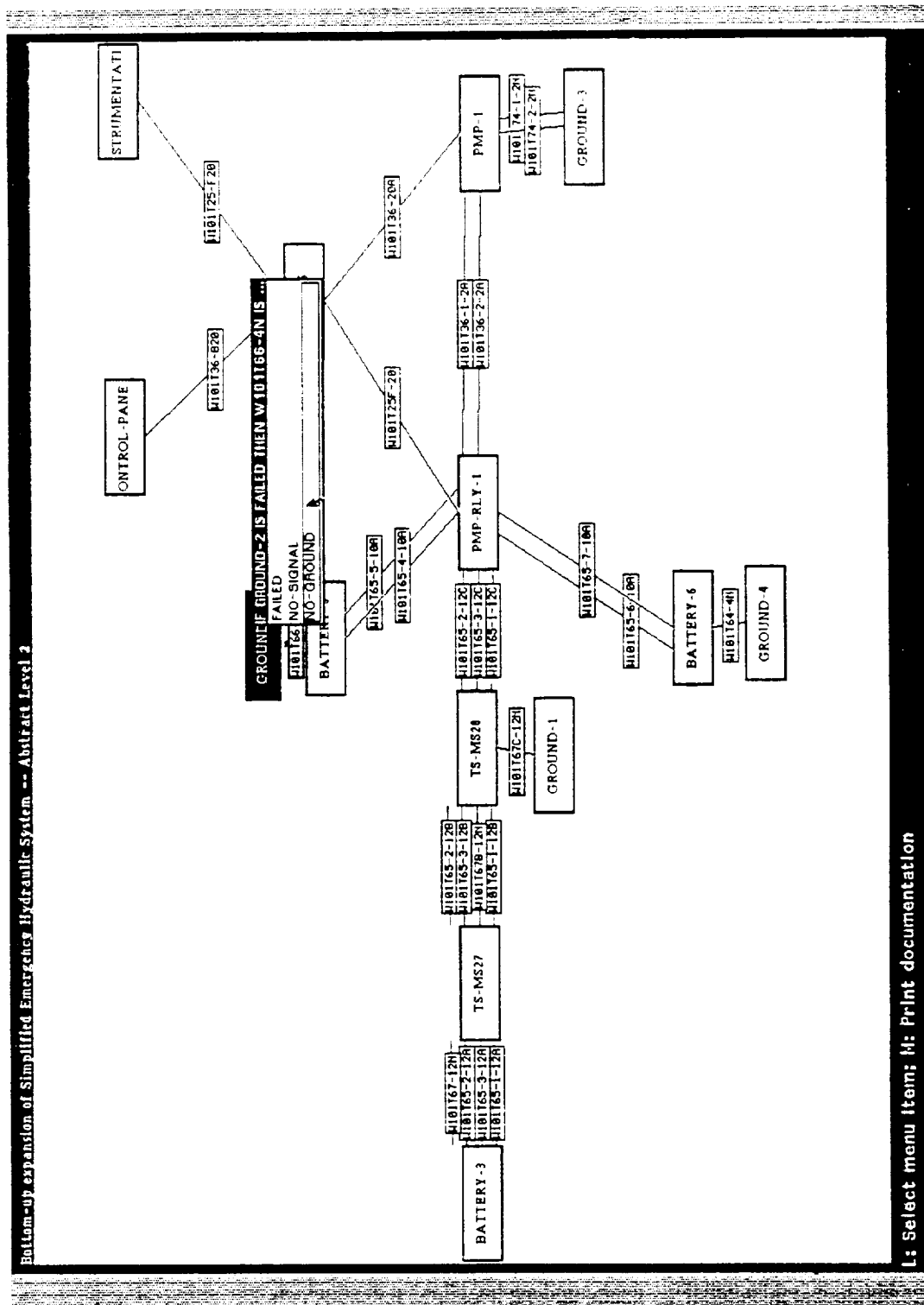


Figure 5-30. Second Bottom Up Abstraction Diagram for Emergency Hydraulic System Model - Level 0



**Figure 5-31. A Typical Rule Capture User Assistance Mechanism**

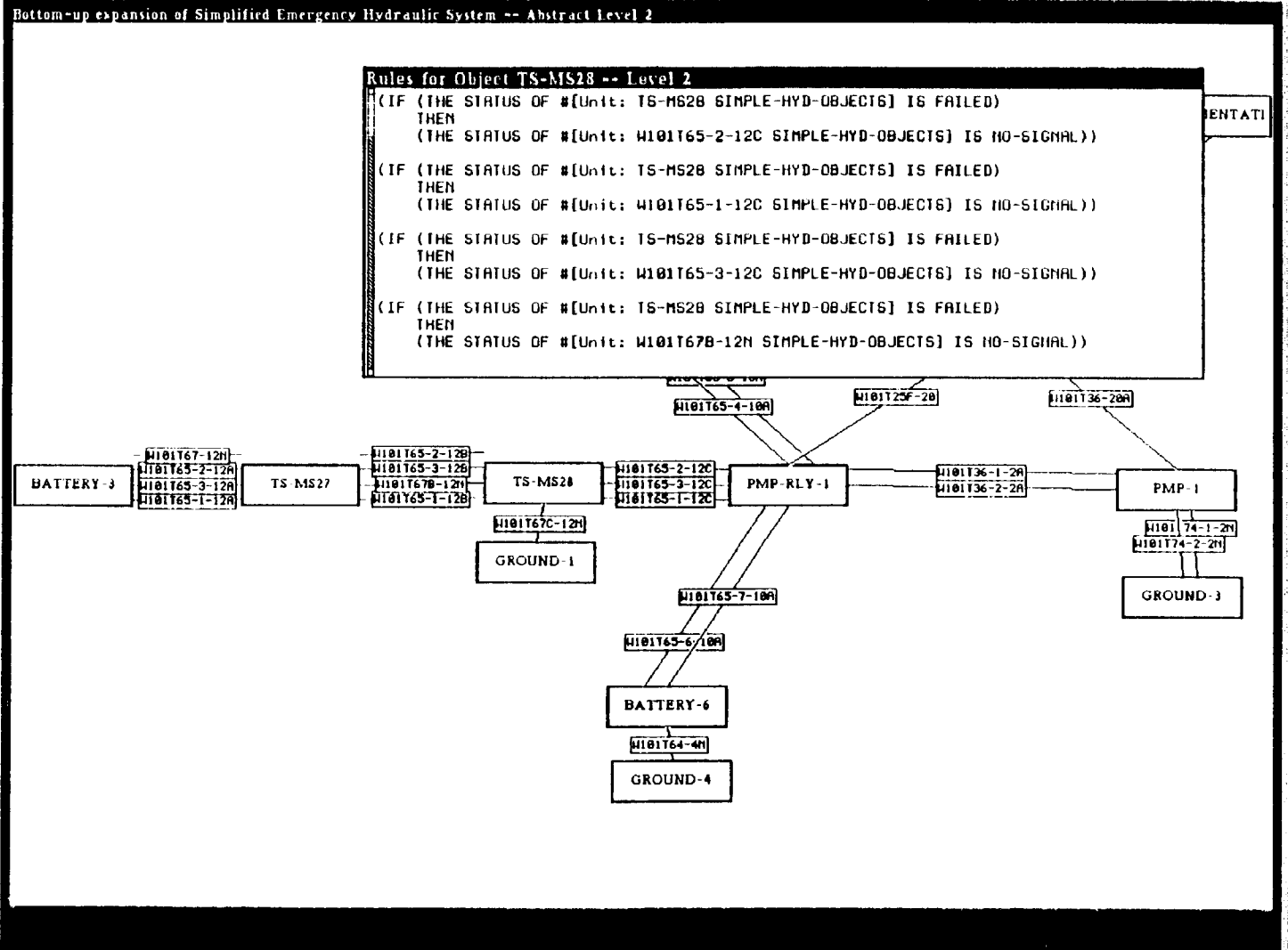


Figure 5-32. A Typical Display of Rules for the EHS Model - These Rules are for a Terminal Strip Failure

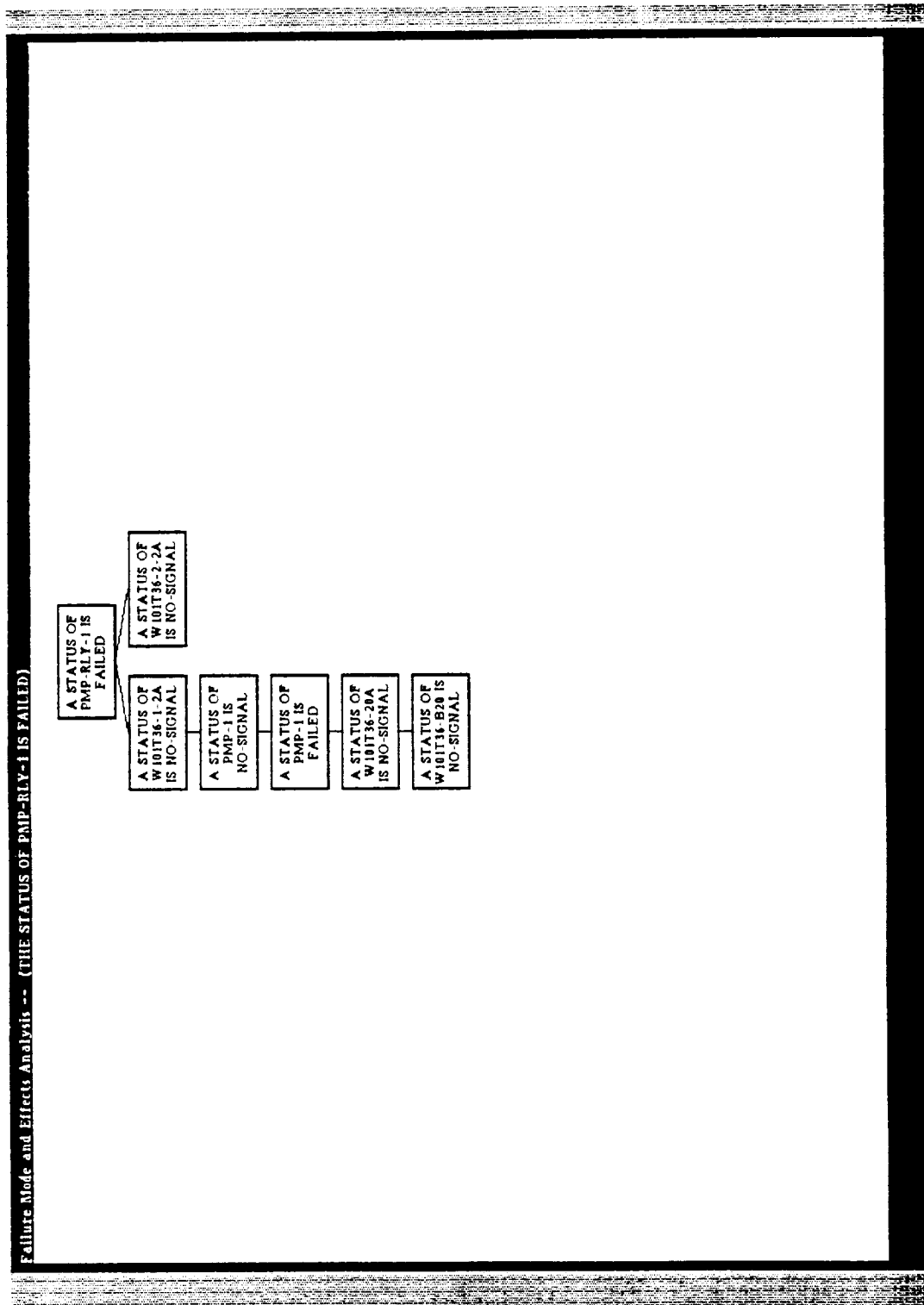


Figure 5-33. An FMEA Display for the Failure of the Pump Relay in the EHS Model

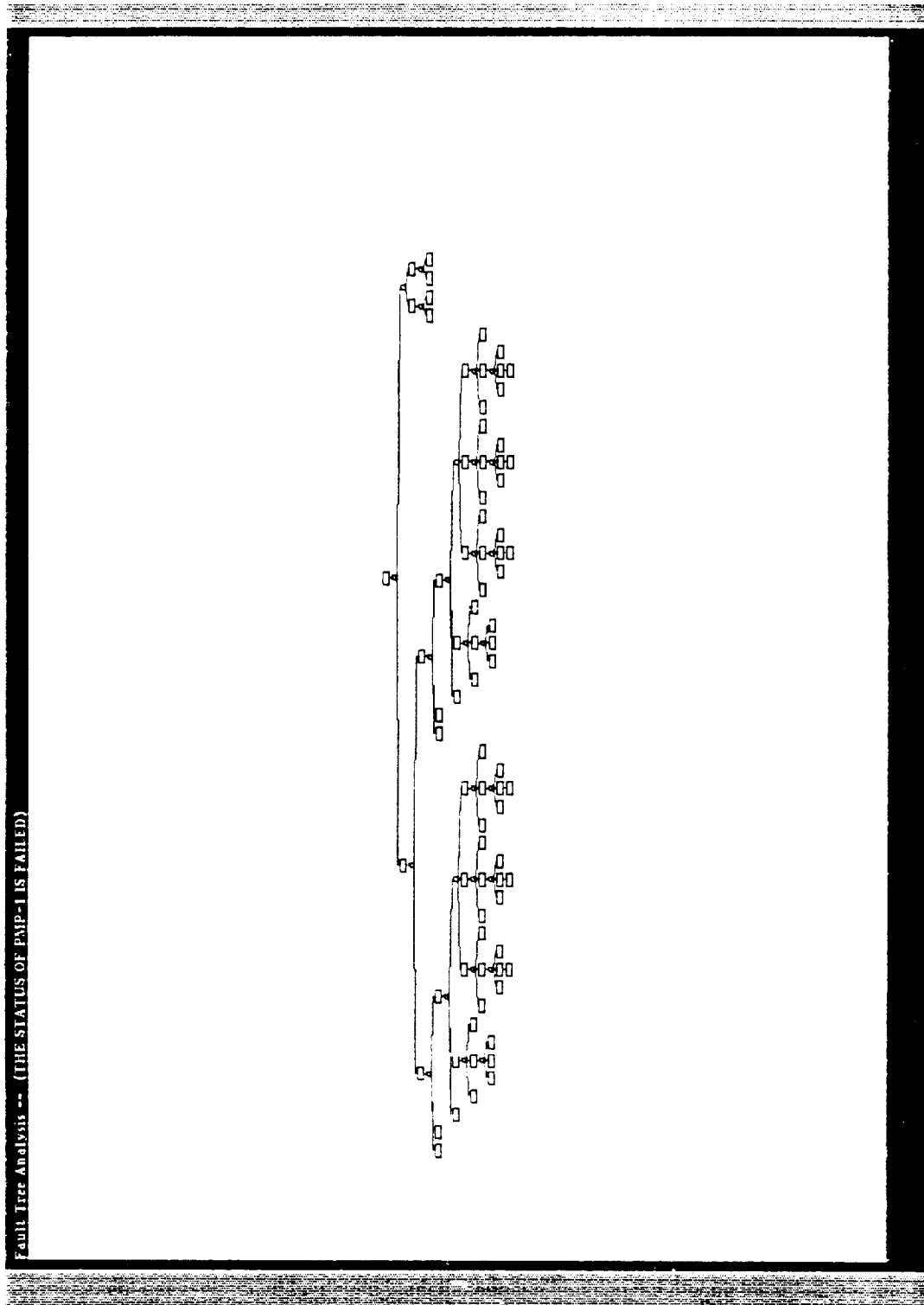


Figure 5-34. An Overview Display of the FTA for the Failure of the Pump in the EHS Model

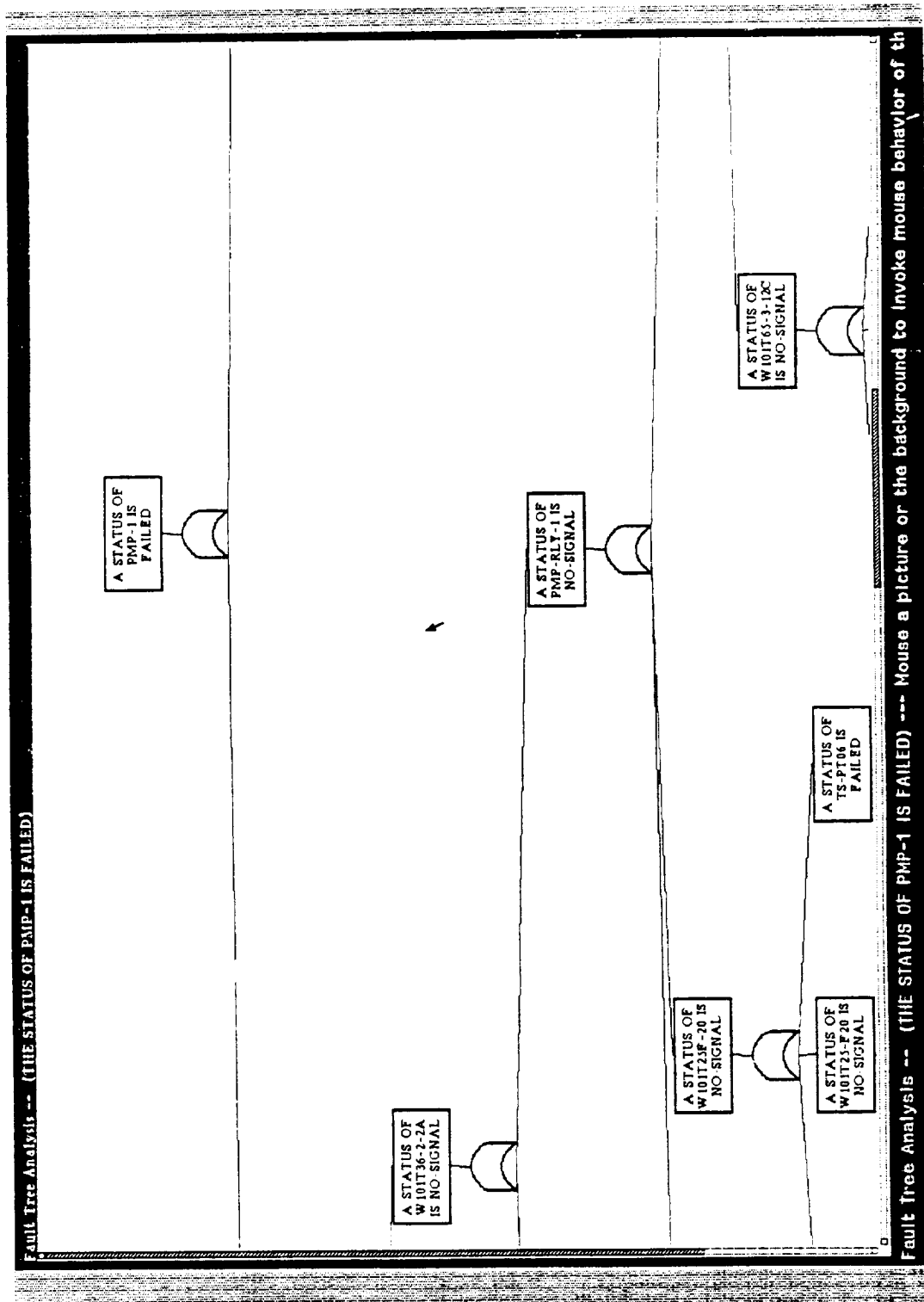


Figure 5-35. An Expanded Display of the FTA for the Failure of the Pump in the EHS Model

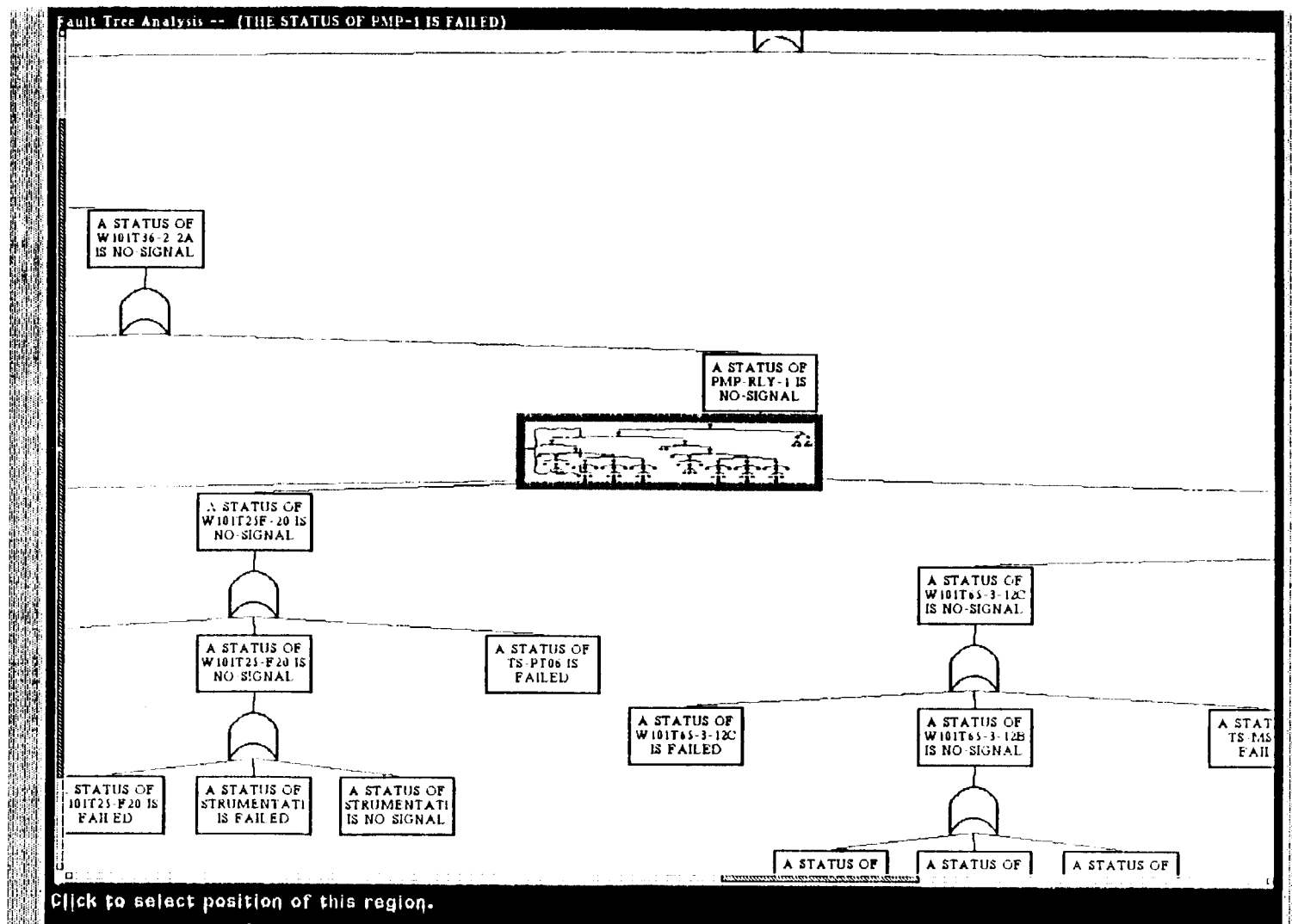


Figure 5-36. An Expanded Display of the FTA for the Failure of the Pump in the EHS Model - With a Browsing Mechanism



## Concluding Remarks

### 6.0 Concluding Remarks

The initiation of this project was based on two primary insights. It was recognized that there was a need for support tools which would assist the development work performed on flight test vehicles at the Dryden Flight Research Facility. In addition, it was recognized that knowledge-based system technology, which has emerged from the academic and laboratory research environment, could satisfy some of these needs. The efforts of this project have produced a prototype KBS which confirms the ability of knowledge-based system technology to support the design and construction of such a tool. The proof-of-concept KCS developed by this project provides a unique capability which was previously unavailable.

The principal goal, which has been accomplished, was the creation of a tool which integrates the knowledge associated with the major design disciplines of an automated control system, in particular an FCS. In addition, this tool captures the integrated knowledge in an environment with an inference engine. Thus, this knowledge can be used with rule-based dynamic models.

#### **6.1 KCS Rationale**

The KCS is designed to enhance the productivity and performance of the design effort associated with an automated system. In addition, it is designed to enhance the operational performance of the automated system in all the subsequent phases of the automated system's life cycle. This life long value, which extends well beyond the design stage, can be used to help justify the use and cost of this KCS. The proof-of-concept system provides evidence that the aforementioned advantages are attainable. The sum total of this evidence is discussed in detail in the sections which elaborate on the KCS overview, knowledge capture concepts, the semantic network and the rule-based models.

As a productivity tool, the KCS supports goals associated with *concurrent engineering* (CE)<sup>8</sup>. CE recognizes that the design and build process consists of multiple disciplines and that these multiple disciplines have been traditionally performed with considerable autonomy, even though the end product depends on a harmonious integration of the multiplicity of disciplinary activities. The efforts of CE are focused on eliminating the inefficiencies that result from the less than harmonious interactions associated with an

## Concluding Remarks

autonomous design and build process. The KCS supports the goals of CE by integrating the knowledge associated with multiple disciplines into a common semantic network. This common semantic network allows the linkages between the multiple disciplines to be identified. In addition, it allows the disparate members of a design team to use a common environment which allows them to better understand the impact of their design decisions on other dimensions of the total system design.

The life-critical nature of the automated systems at the DFRF, such as the HARV FCS, place a high premium on the ability of the automated system to perform properly within its operational envelope. The complexity of a typical system prevents the development of an absolute confidence that the necessary performance standards have been met by the design of the automated system. The KCS should make it possible to develop a higher degree of confidence that the final automated system will work properly within its operational envelope, than that currently possible in a design environment characterized as being autonomous. The visibility into the interaction of the multiplicity of disciplinary design activities provided by the KCS supplies this higher degree of confidence in the design performance.

The browsing and modeling capabilities of the KCS are designed to help the users of a complex automated system understand how that system works. Thus, the same knowledge and interaction between design disciplines that was captured to enhance the performance and productivity in the design phase is available for the operational personnel. This knowledge is supplied via a dynamic, graphic user interface which is intended to encourage users to develop a deeper and broader understanding of the workings of their operational system and to enhance their operational performance.

## **6.2 Lessons Learned**

The goals of the project were accomplished within the constraints of the four man year funding restriction. These limited resources were focused on the development of the semantic network authoring and browsing mechanisms, introduction of existing design knowledge using bitmap technology and the development of rule-based models. The results of this successful use of KBS technology and bitmap technology form the bulk of the results of this effort. On the other hand, the resource limitations prevented exploratory

### Concluding Remarks

use and integration of CAD, CASE and multimedia tools. These tools are viewed as being a highly necessary part of a successful full scale KCS.

The use of graphical images, which used bitmaps, proved to be a mandatory part of the user interface. These graphics were incorporated as an integral part of the decomposition and schematic capture functionality. However, these bitmaps proved to consume a significant amount of memory and in some cases slowed the system response time to an unacceptable degree. With the availability of the current and projected memory capabilities as evidenced by optical storage devices (e.g. CD-ROM ), the bitmap memory requirements may not prove to impose a significant problem. However, the time required to page these large amounts of storage into the machine RAM for display is unacceptable in the current implementation for large schematics. Here, the emergency system schematic, consisting of several 11"x17" drawings, was scanned into the KCS. The display, zooming and scrolling of these bitmap graphics sometimes required the user to wait for minutes of time, while the work station paged through its memory system.

The conclusion from this experience is that the use of bitmaps must be optimized and restricted. The graphical drawings in the user interface, however, must remain as an integral part of the KCS. A related conclusion is that emphasis should be placed on the use of object oriented drawings. CAD and CASE tools are a natural source for these object oriented drawings.

### **6.3 Guidelines for Future Development**

The KCS, as it exists at the conclusion of this project, is a textbook example of a KBS prototype. This project has utilized its rapid prototyping tools to deliver a functional proof-of-concept KCS which complies with the goals discussed in Section 2.0. However, it should be noted that although the proof-of-concept KCS exemplifies the goals of the project, it is subject to review and subsequent improvement for as a prototype KBS it is neither complete nor is it bullet proof.

Rapid prototyping is a KBS concept that advocates the use of multiple iterations in the development process. Each iteration supplies a set of concrete results, which are used to provide direction for the next phase of development. Here, the results of this project are

## Concluding Remarks

viewed as the results of *Phase I* in a multi-phase development of a bullet proof, industrial quality KCS. Phase I has shown that the ambitious KCS goals are viable.

### **6.3.1 Phase II**

In the Phase II stage of development, the KCS should continue to be recognized as a prototype and as such, the development environment should retain its rapid prototyping character. The underlying rapid prototyping tools have proven their worth and should all be retained for use in Phase II. This rapid prototyping environment should be used to develop a restricted set of bullet proof capabilities and to continue exploratory efforts.

It is suggested that in Phase II, the KCS be applied to a small or moderately sized automatic control system within the DFRF flight test environment. This application would provide a stress test of the KCS to guide its development and would simultaneously service known needs within the DFRF flight test environment. Such an application dictates the need for bullet proof KCS capabilities. This development of bullet proof capabilities should focus on the kernel of the KCS which supports authoring and browsing of the semantic network. In addition, effort should be directed toward the optimization and restriction of the use of bitmapped graphics. The graphics should be improved through the use of representative CAD and CASE tools for object oriented graphics as well as the integration of their object definitions into the semantic network. In addition a representative use of multimedia in the user interface should be included.

The Phase II effort also should be used to explore the rich set of possible enhancements which exist for the KCS. The exploratory efforts should include a study of the many CAD, CASE and multimedia tools which may be used to augment the KCS, a meaningful assessment of open environment considerations, a study of the options associated with the transition from a rapid prototyping environment into a full scale application environment, the development of exploratory models which use the inference engine and an investigation of the utility of other dimensions of AI technology.

### **6.3.2 Phase II+**

The end goals of this ambitious project are marked by the obvious need to transition to a bullet proof KCS with industrial strength. This transition is possible in the near term. The

### Concluding Remarks

final version of this NASA development effort is viewed as a technology which is useful to the general business community and appropriate for technology transfer.



## **Bibliography**

1. Minsky, M. L. 1975, *A Framework for Representing Knowledge*. In *The Psychology of Computer Vision*, ed. P. H. Winston, pp. 211-277. New York: McGraw-Hill
2. Buchanan, B and Shortliffe, E. (eds.) 1984, *Rule-Based Expert System: The MYCIN Experiments of the Stanford Heuristic Programming Project*.
3. Shortliffe, E. H. 1974 *MYCIN: A Rule-Based Computer Program for Advising Physicians Regarding Antimicrobial Therapy Selection*. Ph. D. dissertation, Stanford University.
4. Dahl, O.J., Dijkstra, E. W. and Hoare, C.A.R., *Structured Programming*., Hew York: Academic Press, 1972
5. Gan, C., and Sarson T., *Structured Systems Analysis: Tools and Techniques*. New York: Improved System Technologies Inc., 1977
6. Demarco, T., *Structured Analysis and System Specification*, Yourdon Press, 1978
7. Sikora, S., *Modeling Authoring System*, MIT Bachelor of Sciences Thesis, Department of Electrical Engineering and Computer Science, 1990
8. Shumaker, Gerald C. et al, *Initial Program Strategy*, Concurrent Engineering Office, Manufacturing Technology Directorate, Wright Research and Development Center
9. Wechsler, D. B. and Crouse, K. R., *An Approach to Design Knowledge Capture for the Space Station*, NASA TM-89272, 1987
10. Davis, R., *Diagnostic Reasoning Based on Structure and Behavior*, Artificial Intelligence, 24(1):347-410, 1984
11. Hamscher, W., *Model-Based Troubleshooting of Digital Systems*, MIT Artificial Intelligence Laboratory, AI-TR 1074, 1988





1. Report No. NASA CR-4316		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle A Knowledge-Based System Design/Information Tool				5. Report Date August 1990	
				6. Performing Organization Code	
7. Author(s) James G. Allen and Scott E. Sikora (Charles Stark Draper Laboratory, Inc.)				8. Performing Organization Report No. H-1621	
				10. Work Unit No. RTOP 505-68-27	
9. Performing Organization Name and Address Charles Stark Draper Laboratory, Inc. 555 Technology Square Cambridge, MA 02139				11. Contract or Grant No. NAS2-12451	
				13. Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-3191				14. Sponsoring Agency Code	
15. Supplementary Notes NASA Technical Monitor: Dale A. Mackall, NASA Ames Research Center, Dryden Flight Research Facility, Edwards, CA 93523-0273.					
16. Abstract <p>This report discusses the design and application of a proof-of-concept <i>Knowledge Capture System</i> (KCS). The KCS utilizes artificial intelligence (AI) knowledge-based system technology. This KCS has been used to capture selected aspects of the design and implementation of NASA's high angle-of-attack research vehicle (HARV, a modified F/A-18A) flight control system.</p> <p>This KCS is intended for use throughout the FCS life cycle: design, validation, operation, and retrofit. It is intended to minimize design flaws by providing an integrated knowledge base and behavioral models that include safety analyses. The design knowledge captured in the KCS is intended to then be made available to enhance the productivity and performance in all subsequent phases of the FCS life cycle.</p> <p>The KCS consists of a semantic network tailored to conform with the design methodologies associated with structured analysis and conventional H/W design techniques. This semantic network is designed to capture integrated design and implementation information in four disciplines: system design, H/W design, S/W design, and utilities design. Authoring mechanisms are provided to aid the user in capturing the knowledge so as to conform to the structured methodology. These authoring aids support top down decomposition and bottom up aggregation. Browsing mechanisms are provided which are intended to allow the user to peruse this integrated knowledge base. These browsing aids are designed to allow the user to search through the semantic network in a natural fashion.</p>					
17. Key Words (Suggested by Author(s)) Artificial intelligence Flight control Knowledge base Structured analysis				18. Distribution Statement Unclassified-Unlimited  Subject category 66	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 194	
				22. Price A09	





National Aeronautics and  
Space Administration  
Code NTT-4

Washington, D.C.  
20546-0001

Official Business  
Penalty for Private Use, \$300



National Aeronautics and  
Space Administration

Washington, D.C.  
20546

SPECIAL FOURTH CLASS MAIL  
BOOK

Postage and Fees Paid  
National Aeronautics and  
Space Administration  
NASA-451

Official Business  
Penalty for Private Use \$300



L2 001 CR-4316 90080630905694  
NASA  
SCIEN & TECH INFO FACILITY  
ACCESSIONING DEPT  
P O BOX 8757 BWI ARPRT  
BALTIMORE MD 21240

